تطوير صناعة البرمجيات
فى مصر

أبحاث ودراسات

المؤتمر العلمى الخامس لنظم المعلومات وتكنولوجيا الحاسبات
القاهرة: ٩-١١ ديسمبر ١٩٩٧

الذى نظمته

الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات

تحرير

أ. د. محمد محمد الهادى

رئيس الجمعية



الناشر

المكتبة الأكاديمية

٢٠٠٠

تطوير صناعة البرمجيات
في مصـــر

حقوق النشر

المكتبة الاكاديمية

١٢١ شارع التحرير – الدقى – القاهرة

تليفون : ٣٤٨٥٢٨٢ / ٣٤٩١٨٩٠

فاكس : ٣٤٩١٨٩٠ – ٢٠٢

قائمة المحتويات

الصفحة

وقائع أعمال المؤتمر وتوصياته

ملخص أعمال المؤتمر

عقد المؤتمر العلمى الخامس لنظم المعلومات وتكنولوجيا الحاسبات، الذى نظمته الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات تحت موضوع «تطوير صناعة البرمجيات فى مصر Development of Software Industry in Egypt» فى الفترة من ٩-١١ ديسمبر ١٩٩٧ تحت رعاية أ.د. أحمد جويلى، وزير التموين والتجارة.

وقد افتتح المؤتمر يوم الثلاثاء ٩ ديسمبر ١٩٩٧، واشترك فى أعمال المؤتمر حوالى مائة وثمانين عضوا من أساتذة الجامعات والخبراء والمتخصصين والطلاب المهتمين بموضوع المؤتمر المرتبط بتطوير صناعة البرمجيات فى مصر.

وقد هدف المؤتمر دعم صناعة البرمجيات المصرية بإعتبارها من الصناعات الواعدة التى ترتكز على الفكر البشرى الخلاق، وسوف تشكل دعامة للاقتصاد والتنمية القومية فى القرن الحادى والعشرين،، كما سوف تستوعب الآلاف من خريجى الجامعات المصرية للعمل بها سنويا، مما يدعو إلى تشجيع الاستثمارات فيها وجذب الشركات العالمية الرائدة إلى التواجد فى السوق المصرية.

وقد غطى المؤتمر عدة محاور إرتبطت بواقع صناعة البرمجيات فى مصر واتجاهات المستقبل؛ السوق المصرية وصناعة البرمجيات؛ هندسة تطوير البرمجيات من حيث النماذج والأدوات والأساليب؛ معايير ومواصفات تطوير البرمجيات؛ برمجيات الوسائط المتعددة وبرامج التعلم الذاتى والنظم الخبيرة؛ حماية حقوق الملكية الفكرية؛ وتأهيل وتنمية القوى العاملة المتخصصة. ونظمت هذه المحاور فى سبع جلسات عمل، قدم فيها ستة عشر بحثا وعرضا فنيا، وقد خصصت جلستان منها لندوتين ضمتا نخبة من العلماء المتخصصين. وفى إطار المؤتمر نظمت كل من مؤسسة دلتا كمبيوتر والمكتبة الأكاديمية معرضا للبرمجيات والكتب الحديثة المرتبطة بموضوع المؤتمر.

وقد شارك فى حفل افتتاح المؤتمر الأستاذ / سيد أبو القمصان وكيل أول وزارة التموين والتجارة نائبا عن أ.د. أحمد جويلى، وزير التموين والتجارة؛ أ.د. محمد فهمى طلبة، عميد كلية الحاسبات والمعلومات، جامعة عين شمس؛ أ.د. محمد محمد الهادى رئيس المؤتمر؛ وأ.د. علاء الدين محمد فهمى مقرر المؤتمر.

أهداف المؤتمر ومحاوره

عقد هذا المؤتمر لتحقيق الأهداف التالية :

١ ـ وضع معالم استراتيجية وسياسة قومية واضحة لتطوير صناعة البرمجيات ، تحدد الأولويات وتهيئ المناخ الملائم لها .

٢ ـ تأكيد جودة صناعة البرمجيات من خلال الأخذ بالمعايير والمواصفات العالمية وتطويعها للواقع المحلى والعربى والدولى.

٣ ـ خلق نماذج ناجحة لصناعة البرمجيات فى مصر .

٤ ـ دعم البحث والتطوير العلمى لتنمية صناعة البرمجيات .

٥ ـ تشجيع الاستثمار فى صناعة البرمجيات فى مصر .

٦ ـ جذب الشركات الرائدة العالمية فى صناعة البرمجيات إلى التواجد فى السوق المصرية .

وقد نظم المؤتمر فى نطاق المحاور التالية :

١ ـ واقع صناعة البرمجيات فى مصر واتجاهات المستقبل .

٢ ـ اتجاهات الأسواق المحلية والأجنبية لتسويق البرمجيات فى مصر .

٣ ـ عالمية صناعة البرمجيات فى ظل السوق المفتوحة واتفاقية الجات .

٤ ـ الأدوات والأساليب المتقدمة فى تصميم وإنتاج البرمجيات .

٥ ـ معايير ومواصفات صناعة البرمجيات .

٦ ـ تنمية وتأهيل القوى العاملة المؤثرة على صناعة البرمجيات .

٧ ـ التشريعات والقوانين المؤثرة على صناعة البرمجيات

ملخص كلمات جلسة الافتتاح

في إطار جلسة افتتاح المؤتمر حدد الدكتور محمد محمد الهادي، رئيس المؤتمر ورئيس مجلس إدارة الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات، دور المعرفة العلمية وتطبيقاتها في شكل تكنولوجيات متعددة، تكتسب أهمية خاصة في تشكيل معالم الحاضر والمستقبل، إذ انها تعد من أهم مؤشرات التقدم الحضارى. فالمجتمعات المنتجة للمعرفة أقدر على إشباع احتياجاتها من خلال ما تقدمه هذه المعرفة من موارد ضخمة، تصب في الدخل القومي وترفع مستوى معيشة المواطنين. كما أشار إلى أن صناعة البرمجيات في مصر تعد من ضمن الصناعات الإستراتيجية لتميزها بدورة رأس مال سريعة، وتحقيق قيمة مضافة عالية بالمقارنة بالصناعات الأخرى حيث تعتبر القاطرة المحركة للصناعات المتقدمة. كما طالب، رئيس المؤتمر، بضرورة وضع إستراتيجية مصرية لتطوير صناعة البرمجيات في مصر، لكي تكون دعامة التنمية الشاملة في المستقبل، مؤكدا أن مصر لديها العوامل والقدرات الرئيسية التي تؤهلها لأن تمثل مركزا متميزا في تطوير صناعة البرمجيات محليا وإقليميا من خلال توافر القوى البشرية المدربة والمؤهلة على التطوير والإنتاج والتصنيع، وتوافر الحس الاستثمارى العالمي لدى القطاع الخاص بالاستثمار في صناعة البرمجيات، والاستفادة من برامج المشاركة المصرية مع الدول المتقدمة؛ حيث تنعكس آثارها على التصميم والتطوير للبرمجيات المصرية وتسويقها.

وفي جلسة الافتتاح نفسها، أشار الدكتور محمد فهمي طلبة، عميد كلية الحاسبات والمعلومات بجامعة عين شمس، إلى أن سوق البرمجيات بصفة عامة، يتميز بشدة المنافسة مما يتطلب جودة عالية في المنتج، تتواكب مع المتغيرات العالمية السريعة، وتلبي أيضا الاحتياجات المستقبلية. وحدد أنه حينما توضع إستراتيجية لصناعة البرمجيات يجب اعتبار الانتاج للسوق العالمية والإقليمية ولا ننظر للسوق المحلية فقط. كما طالب بضرورة تطوير التشريعات المصرية في هذا المجال لحماية وتشجيع المستثمرين.

وقد أعلن الدكتور أحمد جويلى وزير التموين والتجارة الخارجية فى كلمته التى القاها نيابة عنه الأستاذ سيد أبو القمصان وكيل أول الوزارة لقطاع التجارة الخارجية، أنه سيتم الإعلان عن تشكيل أربعة عشر مجلساً سلعيًّا فى القريب، تنفيذاً لتوصيات مجلس الوزراء الذى طالب بإنشاء هذه المجالس بهدف التنسيق بين الحكومة والقطاع الخاص لزيادة الصادرات، وأن هذه المجالس سوف تشتمل على مجلس لصناعة البرمجيات فى مصر، يمثل فيه كل العاملين فى هذه الصناعة لوضع استراتيجية شاملة لزيادة الصادرات من البرمجيات وحل مشكلات هذا القطاع، وتنمية الكوادر البشرية وبحث المشكلات التسويقية.

وحدد الدكتور جويلى وزير التموين والتجارة الخارجية فى كلمته أن صناعة البرمجيات تمثل عنصرا مهما من عناصر الاقتصاد المصرى، خاصة أن مصر تمتلك الكوادر البشرية المؤهلة لتطوير هذه الصناعة، كما يمكن أن يصل حجم الصادرات المصرية من البرمجيات إلى أرقام عالية منها؛ حيث يصل حجم التجارة العالمية فى هذا المجال إلى أكثر من ١٠٠ مليار دولار سنويا.

وقد أنهى كلمات الافتتاح الدكتور علاء الدين محمد فهمى مقرر المؤتمر موضحاً أن صناعة البرمجيات يمكن أن تضيف - الكثير إلى - الاقتصاد القومى الذى يحظى حاليا بالاهتمام لدى الأوساط العالمية نتيجة الجهود التى انتهجتها الحكومة فى برنامج الإصلاح الاقتصادى، تلك الجهود التى ترمى إلى تطوير الاقتصاد وتنشيط الصادرات. كما طالب بضرورة تحديد النقاط الرئيسية بمسار العمل الإقتصادى لدخول القرن الحادى والعشرين، من خلال إعطاء الأولوية القصوى للتعليم والتدريب، وإدخال التكنولوجيا العالمية لمواجهة التطورات السريعة المتلاحقة فى مجال المعلومات والاتصالات التى ستكون صناعة البرمجيات أحد روافدها الأساسية. وقد اختتم الدكتور علاء الدين محمد فهمى كلمته باستعراض برنامج المؤتمر وتوقيتات جلساته المختلفة.

البرنامج الفعلى للمؤتمر

الثلاثاء : ٩/١٢/١٩٩٧

التسجيل: الساعة ٩,٠٠-١٠,٠٠ صباحا

الإفتتاح : الساعة ١٠,٠٠-١١,٠٠ صباحا

* قراءة آيات من القرآن الكريم
* أ.د. علاء الدين محمد فهمى، مقرر المؤتمر
* أ.د. محمد محمد الهادى، رئيس المؤتمر
* أ.د. محمد فهمى طلبة، عميد كلية الحاسبات والمعلومات، جامعة عين شمس
* أ.د. أحمد جويلى، وزير التموين والتجارة، ألقاها بالإنابة عن سيادته الأستاذ/ سيد أبو القمصان، وكيل أول وزارة التموين والتجارة.

استراحة : الساعة ١١,٠٠ - ١١,٣٠ صباحا

الجلسة الأولى: الساعة ١١,٣٠ - ١,٣٠ ظهرا

«ندوة عن واقع صناعة البرمجيات فى مصر واتجاهات المستقبل»

المقرر : أ.د. محمد محمد الهادى، رئيس المؤتمر

الأعضاء :

* أ.د. محمد فهمى طلبة، عميد كلية الحاسبات والمعلومات، جامعة عين شمس
* أ.د. مصطفى كامل، مدير برنامج التنمية التكنولوجية، مركز المعلومات ودعم اتخاذ القرار برئاسة مجلس الوزراء
* أ.د. نادية حامد حجازى، أستاذ ونائب رئيس المعهد القومى للإلكترونيات
* أ.د. علاء الدين محمد فهمى، مقرر المؤتمر

الأربعاء: ١٠ / ١٢ / ١٩٩٧

الجلسة الثانية: الساعة ٩,٣٠ ـ ١٠,٣٠ صباحا

«السوق المصرية وصناعة البرمجيات»

مقرر الجلسة: أ.د. محمد محمد الهادى، رئيس المؤتمر

المتحدثون:

* أ.د. علاء الدين محمد فهمي، المدير التنفيذى لمؤسسة دلتا للكمبيوتر

"Software Industry in Egypt: Opportunities and Challenges"

* أ.د. نشأت الخميسي محمد الغيطاني، مستشار الحاسبات الآلية ونظم المعلومات

"Software Industry in Egypt: The Current Status and the Ways Towards Promotion"

* أ.د. مصطفى حسام الدين، أستاذ مساعد بقسم المكتبات والمعلومات، كلية الآداب، جامعة القاهرة «الضبط الببليوجرافى للبرمجيات فى جمهورية مصر العربية»

استراحة: الساعة ١٠,٣٠ ـ ١١,٠٠ صباحا

الجلسة الثالثة: الساعة ١١,٠٠ ـ ١,٠٠ ظهرا

«هندسة تطوير البرمجيات: النماذج والأدوات والأساليب»

المتحدثون:

* أ.د. محمد بدر سنوسى، مستشار الحاسبات الآلية ونظم المعلومات

"Abstract Models for an Information System: A First Step to Automate The Analysis Phase"

* أ.د. علاء الدين محمد فهمي، المدير التنفيذى، دلتا للكمبيوتر

"Improving Software Reliability Through Integrated Clean Room and Object Orientation Development"

* د. محمد منير عيسى، د. أحمد طه شهاب الدين

"Towards Improvement of Software Products for Parallel Distributed Systems"

* أ.د. محمد محمد الهادى، أستاذ نظم المعلومات المتفرغ، أكاديمية السادات للعلوم الإدارية

"Technological Implications on Software Development"

استراحة : الساعة ١,٠٠ - ١,٣٠ ظهراً

الجلسة الرابعة: الساعة ١,٣٠ - ٣,٠٠ عصراً

«معايير ومواصفات جودة تطوير البرمجيات»

رئيس الجلسة : أ.د. عبدالبديع محمد سالم «أستاذ ووكيل كلية الحاسبات والمعلومات، جامعة عين شمس»

المتحدثون :

* أ.د. علاء الدين محمد فهمى، المدير التنفيذى لمؤسسة دلتا للكمبيوتر

"An Approach for Evaluation of Software Quality"

* أ.د. نشأت الخميسى محمد الغيطانى، مستشار الحاسبات الآلية ونظم المعلومات

"An Analytical Evaluation of the Impact of Computer Technology on Managerial Levels"

* د. محمد منير عيسى، د. فرحات فرج فرحات، أكاديمية السادات للعلوم الإدارية

"Multimedia Software Development Standards and Protocols"

الخميس : ١١/ ١٢/ ١٩٩٧

الجلسة الخامسة : الساعة ٩,٠٠ - ١١,٠٠ صباحاً

«برمجيات الوسائط المتعددة، وبرامج التعلم الذاتى والنظم الخبيرة»

رئيس الجلسة : أ.د. سيد محمد عبدالوهاب، أستاذ ورئيس قسم الحاسب الآلى ونظم المعلومات، أكاديمية السادات للعلوم الإدارية

المتحدثون :

* أ.د. عايدة عبداس أبوغريب، د. شعبان حامد على إبراهيم

المركز القومى للبحوث التربوية والتنمية

«برامج إثرائية للتعلم الذاتى فى مناهج المرحلة الابتدائية باستخدام الوسائط المتعددة»

* د. محمد منير عيسى، د. فرحات فرج فرحات، أكاديمية السادات للعلوم الإدارية

"Hypermedia Software Development Technologies"

* د. مجدى أبو العلا، م. تامر محمد، م. ياسر رمضان

"Hyprid Statutory Reasoning and Information Retrieval Systems"

* د. عطا إبراهيم الألفى، كلية التربية النوعية بالمنصورة

إنتاج برمجيات الدليل القومى للتربية التعاونية بمصر «دراسة حالة»

استراحة: الساعة ١١,٠٠ - ١١,٣٠ صباحا

«الأوجه القانونية لصناعة البرمجيات في مصر»

رئيس الجلسة: أ.د. أحمد شرف الدين، أستاذ القانون المدنى، كلية حقوق عين شمس

المتحدثون:

* عزة محمود خليل، المحكم الدولى بمركز القاهرة للتحكيم التجارى الدولى والمحامية بالنقض

«المسئولية القانونية الناشئة عن استخدام البرامج على شبكة الإنترنت»

* أ.د. محمود غنيم، أستاذ قانون الاتصالات والفضاء الخارجى، جامعة ماكجيل، مونتريال، كندا

«القانون الدولى لحماية الإتصالات والفضاء الخارجى وتدفق البرمجيات والمعلومات»

إستراحة: الساعة ١,٠٠ - ١,٣٠ ظهرا

الجلسة السابعة: الساعة ١,٣٠ - ٣,٠٠ عصرا

«ندوة تأهيل وتنمية القوى العاملة المتخصص»

مقرر الجلسة: أ.د. محمد عبد الحليم، أستاذ ووكيل كلية التربية، جامعة حلوان

المشتركون:

* أ.د. مصطفى بهجت عبد المتعال، أستاذ ورئيس جامعة المنوفية الأسبق

* أ.د. سيد محمد عبد الوهاب، رئيس قسم الحاسب الآلى ونظم المعلومات بأكاديمية السادات

* أ.د. محمد محمد فهمى طلبة، عميد كلية الحاسبات والمعلومات، جامعة عين شمس

* أ.د. محمد محمد الهادى، أستاذ نظم المعلومات المتفرغ بأكاديمية السادات

جلسة الختام والتوصيات: الساعة ٣,٠٠ - ٣,٣٠ ظهرا

* أ.د. محمد محمد الهادى، رئيس المؤتمر

* أ.د. علاء الدين محمد فهمى، مقرر المؤتمر

التوصيات

(١) اعتبار صناعة البرمجيات صناعة إستراتيجية حاكمة، تؤثر على القوة النسبية للدولة وأمنها القومى مما يتطلب تنسيق جميع الجهود الحكومية وغير الحكومية فى جميع قطاعات الدولة على كافة مستوياتها وتوجهاتها؛ لوضع تلك الصناعة فى إطارها الصحيح الذى يتناسب مع أهميتها القصوى، مع قيام حملات توعية فى وسائل الإعلام المختلفة التى تنبه لذلك.

(٢) ضرورة تبنى إستراتيجيات وخطط وبرامج قومية لنقل التكنولوجيا المتقدمة لقطاع صناعة البرمجيات المصرية.

(٣) تشجيع القيام بالصناعات الصغيرة للقوى البشرية المؤهلة لبدء صناعة البرمجيات ذات الطابع الفكرى الإبداعى.

(٤) ضرورة القيام بدراسة الأسواق المحلية والعربية والأجنبية من حيث العرض والطلب على البرمجيات، والعمل على توفير البيانات والإحصاءات الدقيقة عن هذه الصناعة للشركات والأفراد المهتمين.

(٥) وضع الوسائل والآليات التى تحقق سهولة مشاركة الشركات والأفراد المطورين للبرمجيات المصرية فى الجهود العالمية لتقديم حلول لمشكلة عام ٢٠٠٠، لدى الشركات والهيئات على كافة المستويات المحلية والإقليمية والدولية.

(٦) إعطاء الأولوية لإنتاج البرمجيات ذات التوجه الإسلامى واستثمار السوق الكبير على إمتداد العالم الإسلامى والأقليات الإسلامية فى العالم الغربى؛ للاستفادة من الميزة النسبية لمصر فى ملكية القطاع الأعظم من كتب التراث والمؤلفات المعاصرة والسبق فى مجال التعريب والترجمة إلى لغات أخرى.

(٧) إعطاء ميزة نسبية للبرمجيات المصرية فى المناقصات العامة، وتقديم حافز تفضيلى للمنتج المحلى ووضع الضوابط التى تحقق تعظيم الطلب على البرمجيات المصرية.

(٨) توجيه آلية البحث العلمي والتطوير لإنتاج أدوات وأساليب وتكنولوجيات جديدة، تسهم فى هندسة تطوير صناعة البرمجيات فى مصر.

(٩) التعاون مع الهيئة العامة للتوحيد القياسى الخاصة بجودة البرامج واختبارها وإصدار شهادات الصلاحية لها.

(١٠) وضع صناعة البرمجيات المصرية فى أولويات الصناعات المطلوب الحصول لها على تمويل من المؤسسات المالية والمصرفية بشروط ميسرة، وتوفير نظم ائتمانية لها تساير طبيعة هذه الصناعة الفكرية، التى لا تناسبها القواعد الائتمانية التقليدية.

(١١) إعفاء كافة أعمال البرمجيات من الضريبة على المبيعات حيث لا ينصب هذا النشاط على سلعة فهى إنتاج فكرى إلى حد كبير لا يترتب عليه تحويل أى مادة إلى منتج جديد أو تغيير مواصفاتها، وذلك إسوة بالإعفاءات التى تتمتع بها الكتب.

(١٢) الإستفادة من برامج الشراكة المصرية القائمة مع الدول المتقدمة بحيث تنعكس آثارها على دعم صناعة البرمجيات المصرية وفتح منافذ تسويقية خارجية لها.

(١٣) تنظيم المهن المرتبطة بصناعة البرمجيات ووضع مواصفاتها وتحديد المهارات والكفايات اللازمة لأدائها مع رعاية العاملين بهذه المهن فى نطاق ضوابط وميثاق شرف مهنى، يحدد أخلاقيات وآداب الانتماء لها.

(١٤) توفير وتنمية القوى العاملة المتخصصة، التى تستطيع مواكبة التطوير السريع المتلاحق فى هذه المناهج وضرورة ربط المعاهد والجهات العلمية بشركات صناعة البرمجيات لتطوير المناهج، بما يتناسب مع احتياجات السوق لمنتجات البرمجيات واستخدماتها، من خلال:

أ . إنشاء مركز قومى لتطوير مقررات ومناهج علوم الحاسبات والمعلومات والاتصالات والإلكترونيات المرتبطة بصناعة البرمجيات.

ب. تطوير مقررات ومواد دراسية لعلوم الحاسبات والمعلومات تدرس فى مراحل التعليم قبل الجامعى كركائز، تبنى عليها مقررات التعليم الجامعى والعالى.

ج. تطوير مقررات ومناهج دراسية لعلوم الحاسبات والمعلومات لتأهيل وإعداد أخصائيى الحاسبات والمعلومات فى مجالات التطبيقات المختلفة والمتنوعة.

د . التأكيد على ضرورة توفير هيئات تدريس وتدريب كافية ومؤهلة على أعلى مستوى ترتبط بجودة العملية التعليمية والتدريبية، مع إعادة تدريبها بصفة مستمرة.

هـ. التوسع في التدريب التحويلي للقوى البشرية العاملة لكي تتخصص في صناعة البرمجيات واستخداماتها.

(١٥) ضرورة إعادة المفهوم الذى بنيت عليه القوانين والتشريعات القائمة من حيث:

أ . حماية الملكية الفكرية لمنتجات البرمجيات.

ب . إيداع نسخ من البرمجيات المنتجة في مصر في الهيئة القومية لدار الكتب والوثائق، وفقاً لقانون الإيداع القانوني أسوة بالكتب والمطبوعات.

ج . إصدار الببليوجرافيات والأدلة التي تعرف بالبرمجيات المنتجة في مصر أو عنها والشركات المنتجة لها.

د . دعم براءات الاختراع لإنتاج أدوات وأساليب وتكنولوجيات تعمل على تطوير البرمجيات.

هـ . التنسيق دوليا لوضع التشريعات التي تحقق حماية كاملة للبرمجيات على شبكات المعلومات العالمية كشبكة الإنترنت وأسلوب تداولها خلال الشبكات.

شكر وتقدير

تتقدم الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات والسادة الأعضاء المشتركين في المؤتمر بالشكر والتقدير للجهات التالية، التي ساهمت ماديا ومعنويا في نجاح المؤتمر؛ وهي:

(١) الأستاذ الدكتور/ أحمد جويلي، وزير التموين والتجارة لموافقته على وضع المؤتمر تحت رعايته، وإنابة الأستاذ/ سيد أبو القمصان وكيل أول الوزارة بافتتاحه.

(٢) الأستاذ الدكتور/ حمدى عبد العزيز، رئيس أكاديمية البحث العلمي والتكنولوجيا على الدعم المادى الذى قدم لعقد المؤتمر.

(٣) الأستاذ المهندس/ عمرو علوبة، مدير شركة تى كمبيوتر على الدعم الذى قدمه للمؤتمر ومجلة الجمعية.

(٤) الأستاذ المهندس/ إسماعيل عثمان، رئيس شركة المقاولون العرب، والأستاذ/ حمدى إدريس، مدير الأمن بها على الدعم المقدم للمؤتمر من خلال الترتيبات التمهيدية وطباعة أوراقه ووثائقه.

(٥) الأستاذ الدكتور/ أحمد محمد السعيد، مدير عام مركز الأهرام وتكنولوجيا المعلومات بمؤسسة الأهرام على إعداد اللوحات التعريفية للمؤتمر، وكل ما يتعلق بالخطوط المتعلقة بترتيباته وحضوره.

(٦) الأستاذ الدكتور علاء الدين محمد فهمي، المدير التنفيذى لمؤسسة دلتا كمبيوتر على إقامتهم معرضا وتقديمهم عونا ماديا للمؤتمر.

(٧) الأستاذ/ أحمد أمين، مدير المكتبة الأكاديمية على تنظيم معرض، أثناء فترة عقد المؤتمر ضم إصدارات مؤتمرات الجمعية السابقة.

كلمة الافتتاح الرئيسية

أ. د. محمد محمد الهادى

الإخوة الحضور، حضرات السادة والسيدات، الأستاذ سيد أبو القمصان وكيل أول وزارة التجارة الخارجية مندوبا عن الدكتور أحمد جويلى وزير التموين والتجارة الخارجية، الأخ الدكتور محمد فهمى طلبة، عميد كلية الحاسبات والمعلومات بجامعة عين شمس. سلام الله عليكم ورحمته وبركاته،

شكرا لكم جميعا على تكرمكم بالمشاركة والحضور إلى المؤتمر العلمى الخامس لنظم المعلومات وتكنولوجيا الحاسبات، الذى تنظمه وتعقده الجمعية المصرية لنظم المعلومات وتكنولوجيا الحاسبات تحت موضوع «تطوير صناعة البرمجيات فى مصر» الذى تفضل أ.د. أحمد جويلى وزير التموين والتجارة الخارجية بوضعه تحت رعايته، ولكن حالت ظروف قاهرة بأن يكون معنا اليوم.

اسمحوا لى أن أبدأ كلمتى بأن هناك إجماعاً فى الرأى على أن إنتاج المعرفة العلمية وتطبيقاتها فى شكل تكنولوجيات منها تكنولوجيا صناعة البرمجيات يعد من أهم مؤشرات التقدم المعاصر؛ أى إن المجتمعات المنتجة للمعرفة أقدر على إشباع الحاجات الأساسية للمواطنين، بالإضافة إلى ما تدره هذه المعرفة العلمية والتكنولوجية من موارد ضخمة تصب فى الدخل القومى، وتعود بالنفع على المواطنين، فى حين أن المجتمعات المستهلكة للمعرفة وتطبيقاتها تبقى فى وضع تابع علميا وتكنولوجيا بكل ما يعنيه ذلك من تأثير على القوة النسبية للدولة وأمنها القومى، بالإضافة إلى التكاليف الباهظة التى يتحملها الاقتصاد القومى فى مجال نقل التكنولوجيا، وعدم ملاءمتها فى كثير من الأحيان للظروف المحلية.

والأهمية الفائقة فى إنتاج المعرفة العلمية تتمثل فى أننا نعيش حاليا فى عصر ثورة المعلومات والمعرفة، حيث يمكن بالعلم والتكنولوجيا حل كثير من المشكلات الإنسانية والمساعدة فى تحسين جودة الحياة التى نعيشها.

هذه الحقيقة مثلت القضية الأولى التى أثارها السيد رئيس الجمهورية فى منتدى «دافوس» الاقتصادى بسويسرا فى شهر فبراير ١٩٩٧، حيث وصفها بأنها أول المرتكزات للنظام

العالمى فى نهاية القرن العشرين، وهى التى أدت إلى ظاهرة العولمة التى تطرق إليها السيد رئيس الجمهورية.

كما أكدها أيضا الرئيس مبارك فى خطابه أمام الاجتماع المشترك لمجلسى الشعب والشورى الأخير بضرورة إدخال التكنولوجيا الرفيعة أو العالمية، التى تمثل إحدى البنيات الأساسية لدخول القرن الحادى والعشرين، لذلك يصبح الاستثمار فيها ذا أولوية قصوى.

من هذا المنطلق كان اختيارنا لموضوع هذا المؤتمر «تطوير صناعة البرمجيات فى مصر»، حيث إنها صناعة إستراتيجية ذات دورة رأس مال سريع، كما أنها صناعة ذات قيمة مضافة عالية بالمقارنة بصناعات أخرى، كما أن العقل البشرى هو الثروة الحقيقية لهذه الصناعة. فصناعة البرمجيات تعتبر القاطرة التى تدفع الصناعات التكنولوجية المتقدمة بصفة خاصة وغيرها من الصناعات الأخرى بصفة عامة، حيث تمثل السبيل الرئيسى لتشغيل الحاسبات ودعامة للاتصالات وشبكات المعلومات ومنطلقا لإطلاق الصواريخ ودوران الأقمار الصناعية والإدارة والخدمات. ويترتب على ذلك أن الدول التى تنأى بنفسها عن الدخول فى صناعة البرمجيات العقلية تهيئ نفسها لمعدلات عالية من الاستيراد لهذه البرمجيات، وما يمثله ذلك من عبء واضح على اقتصادها الوطنى.

وهناك دول أدركت أهمية صناعة البرمجيات، وأعطتها أولوية قصوى على أعلى مستويات اتخاذ القرار بها كما حدث فى الهند وماليزيا والصين، وأخيرا مصر كما سبق الإشارة إليه فى توجهات رئيس الجمهورية الأخيرة، وكما تمثل فى القرار الجمهورى الصادر فى ٢٠ أغسطس ١٩٩٧ بتعديل الجمارك على برامج وأنظمة تشغيل الحاسبات الآلية وبرامج تطبيقها وتخفيض الجمارك عليها من ٣٠٪ إلى ٥٪ فقط؛ مما كان له انطباع إيجابى لدى منتجى البرمجيات العالمية والمحلية على حد سواء، وكما قررته لجنة الاتصالات بالسماح لمقدمى خدمات الإنترنت بممارسة أعمالهم، دون رسوم ضريبية للترخيص خلال العاملين الأولين من بدء النشاط بهدف دعم نمو الخدمات وتحميل البرمجيات على الإنترنت.

أى أن مصر تسعى مثلها مثل كثير من الدول النامية والتى فى سبيل التقدم؛ خاصة المتواجدة فى شرق آسيا إلى أن يكون لها نصيب فى صناعة البرمجيات الدولية عن طريق دعمها للاستثمارات وتشجيعها.

بل إن دولة كالهند على سبيل المثال حققت صناعة البرمجيات لها العام الماضى (١٩٩٦) فقط صادرات بلغت ما يقرب من أربعة مليارات دولار أمريكى، بينما لا تتجاوز صادرات

مصر من البرامج فى العام الماضى ٥ ملايين دولار فقط، كما أننا نلاحظ أن الدول المتقدمة كالولايات المتحدة الأمريكية واليابان والإتحاد الأوروبى بل وإسرائيل تزداد منتجاتها من البرمجيات المتقدمة، التى تستوعبها أسواقها وتصدر إلى الدول النامية التى من بينها مصر بطبيعة الحال.

إن صناعة البرمجيات تعتمد على توجهين أساسيين هما: الإبتكار أو الإبداع، والدقة. حيث إن الابتكار والقدرة على الاختراع والإبداع والدقة فى التطوير والإنتاج والتصنيع يمثلان عقل صناعة البرمجيات التى تطوعها وتشكلها الإبتكارات الجديدة مما يعطيها المستوى العالمى المرتفع جوهر هذه الصناعة التى تقوم بتحويل الأفكار إلى منجزات تتميز بالدقة والكفاءة الهائلة.

وتتوفر قوى رئيسية تؤثر على صناعة البرمجيات العالمية حالياً، منها ما يلى:

١ ـ النسبة الكبيرة فى تكاليف تطوير البرمجيات ترتبط بتكاليف إنتاجها.

٢ ـ ازدياد الجهود الدولية فى تقنين نظم التشغيل OS.

٣ ـ الابتعاد عن الحلول المرتبطة بالمورد الواحد، والتوجه نحو الحلول المتكاملة المعتمدة على موردين عديدين.

٤ ـ زيادة التركيز على إنتاج البرمجيات وتسويقها بواسطة موردى الأجهزة.

٥ ـ التوسع فى صناعة البرمجيات أدى إلى تواجد موردين عددين لها.

إن أى رؤية أو استراتيجية منشودة لصناعة البرمجيات المصرية يجب أن تتميز بالتالى:

(١) تعظيم القيمة المضافة على نطاق الاقتصاد الوطنى، من خلال التوجه المستمر للتصنيع المصرى والاعتماد على آليات العرض والطلب.

(٢) الاهتمام بالصناعات الصغيرة أو الفردية للقوى البشرية المؤهلة لبدء صناعتها الفعلية.

(٣) الإنتاج الجيد والتسويق الذى يعتبر المدخل الرئيسى للدخول فى رحاب المستقبل.

(٤) التوسع فى التدريب التحويلى لهذه الصناعة الفكرية.

(٥) تطوير مكونات متميزة لمعالجة النص العربى المكتوب والمنطوق وتطويعه إلكترونياً للحاسبات الآلية.

(٦) دمج مكونات الصناعة المصرية داخل النظم الدولية بحيث تترجم المعالجات الذكية الداخلية إلى تطبيقات عملية تخدم المستخدم العربى، وتتيح له إمكانية التعامل مع الحاسبات الآلية وشبكات المعلومات كالإنترنت.

أى يجب توفير عناصر أساسية لتدعيم تطوير هذه الصناعة والتى تتمثل فى التالى:

- القوى البشرية القادرة على التصنيع والإنتاج.
- رأس المال ذو التوجه الإنتاجى الصناعى للفكر.
- توفير الأسواق القادرة على استيعاب المنتج الصناعى.
- توفر المناخ العام المناسب والمشجع على الصناعة.

على أننا نلاحظ وجود كثير من المشكلات، التى تحد من تطوير صناعة البرمجيات فى مصر التى منها:

١ - نقص الخبرة والمعرفة الفنية لمطورى البرمجيات المحللين.

٢ - صعوبة قياس جهود تطوير البرمجيات بطرق مناسبة.

٣ - ارتفاع تكاليف دورة حياة تطوير البرمجيات وعدم تصميمها من أجل إعادة الاستخدام Reusability

٤ - تنفيذ التصميمات وتطبيقها قبل ظهور جودتها.

٥ - افتقاد نظم إدارة الجودة الشاملة TQM فى صناعة البرمجيات المصرية.

٦ - غياب التخطيط للنهوض بالصناعة المحلية.

٧ - عدم مراعاة الأبعاد البيئية المتنوعة فى التطبيق.

حضرات السادة والسيدات، لا يمكننا ونحن على مشارف القرن الحادى والعشرين تجاهل التقدم المذهل فى تقنيات صناعة البرمجيات والحاجة إلى وضع إستراتيجية مصرية لتطوير هذه الصناعة الفكرية، التى سوف تكون دعامة التنمية الشاملة فى المستقبل. إن عدم وجود إستراتيجية محددة وواضحة لصناعة البرمجيات فى مصر حتى الآن أدى إلى عدم وضوح الدور، الذى يمكن أن تلعبه مصر فى هذا المضمار والأسلوب الذى يجب أن ينتهج للقيام

بذلك، والتغلب على المشكلات السابق الإشارة إليها. إن غياب هذه الاستراتيجية أو الرؤية هو السبب الرئيسى وراء عدم تقدمها، ولكنها تمثل البنيات الأساسية المطلوبة لتطوير صناعة البرمجيات فى مصر التى يجب أن ترتكز عليها. وترتبط البنيات الأساسية فى التالى:

أولا ـ القوى البشرية المؤهلة والمعدة: وما يحتاجه ذلك من نهوض تقنى من خلال برامج تدريب مكثفة بالاستعانة بالخبرات المتاحة محليا ودوليا. ويستتبع ذلك تعديل وتطوير وتقويم برامج التعليم الحالية لكى تتناسب مع احتياجات المستقبل، ويواكب ذلك ضرورة توفير خطة لرفع الوعى التكنولوجى العام عن طريق المدارس والجامعات ووسائل الإعلام... ويتمثل ذلك حاليا فى مشروع مبارك لتطوير التعليم فى مصر، الذى نال عنه جائزة اليونسكو وتقديرها.

ثانيا ـ الاستثمار: وما يتطلبه ذلك من تذليل العقبات لإنجاح الصناعات القائمة لتشجيع أنشطة تعمل فى مجال تصنيع البرمجيات، ورفع الأعباء المالية عند بدايات المشروعات وتوفير التمويل المناسب بشروط ميسرة، وتوجيه المنح والقروض ذات الأجل الطويل وبفترات سماح ملائمة، علاوة على إعادة النظر فى قوانين الضرائب والجمارك الراهنة بما لا يمس دخل الدولة.

ثالثا ـ التسويق: وما يمكن عمله لتنشيط متطلبات السوق المحلية عن طريق طرح المتطلبات المرتبطة بتطوير نظم المعلومات على شركات القطاع الخاص، وضرورة توفير الإحصائيات الدقيقة لاحتياجات السوق المحلية والأسواق العالمية..

رابعا ـ التكنولوجيا: توفير كل جديد من تكنولوجيات هندسة تطوير البرمجيات من نماذج وأدوات وأساليب متقدمة والتدريب عليها.

خامسا ـ التشريعات: ضرورة إعادة المفهوم الذى بنيت عليه التشريعات القائمة.

سادسا ـ البحث العلمى: يجب أن يتحول البحث العلمى إلى آلية من آليات التنمية بدلا من قصره على التعليم فحسب، إذ إن التنمية التكنولوجية للبرمجيات تعتمد على البحث العلمى والتطوير.

الإخوة والأخوات، إن مصر يمكن أن يكون لها دور ريادى فى صناعة البرمجيات فى المنطقة العربية وفى العالم من حولنا، إذ يتوفر لها عدة مقومات ترتبط بالبنيات الأساسية السابقة، ومن هذه المقومات ما يلى:

١ ـ توفر القوى العاملة المتخصصة التى تستطيع مواكبة هذه الصناعة.

٢ ـ ازدياد الوعى لدى رجال الأعمال بالاستثمار فى هذه الصناعة الواعدة.

٣ ـ إيجاد التوازن بين الصناعة المحلية واستجلاب الشركات الأجنبية لعولمة السوق.

٤ ـ تعظيم الطلب على البرمجيات داخل مصر من خلال زيادة الاعتمادات المالية للقطاعات الحكومية.

٥ ـ بناء خطط وبرامج قومية لنقل التكنولوجيا المتقدمة لقطاع صناعة البرمجيات فى مصر.

٦ ـ الاستفادة من برامج الشراكة المصرية القائمة مع الدول المتقدمة بحيث تنعكس آثارها على التصميم والتطوير للبرمجيات المصرية وتسويقها.

٧ ـ المشاركة فى الجهود العالمية لتقديم حلول لمشكلة عام ٢٠٠٠ لدى الهيئات والشركات على المستوى المحلى والعالمى، فالاستثمارات المطلوبة لحل هذه المشكلة يمكن أن تتجاوز ٦٠ مليار دولار، تعمل دولة كالهند على أن تستحوذ على ٧ مليارات منها.

حضرات الحضور، هذه بعض الانطباعات عن صناعة البرمجيات وكيفية العمل على تطويرها والنهوض بها لخدمة الاقتصاد والتنمية المصرية مما ينعكس على تقدم مصر وتبوء مكانتها الريادية فى عالم الغد.

شكرا لكم جميعا على حرصكم على الحضور والمشاركة فى أعمال هذا المؤتمر الذى نأمل له النجاح وتحقيق أهداف عقده. وفقنا الله وإياكم لمناقشات وتوصيات مجدية وتطلعا لدعم صناعة البرمجيات والعاملين فيها على كافة مستوياتهم وتوجهاتهم.

والسلام عليكم ورحمة الله وبركاته.

أ. د. محمد محمد الهادى

٩ ديسمبر ١٩٩٧

الجزء الأول

السوق المصرية
وصناعة البرمجيات

صناعة البرمجيات فى مصر
الفرص والتحديات

د. علاء الدين محمد فهمى

المستخلص

إن دور ووضع صناعة البرمجيات فى مصر يعانى من درجة كبيرة من الفهم الخاطئ من العالم الغربى. والأهمية الاقتصادية والسياسية والإستراتيجية لمصر جعلت هذه المفاهيم الخاطئة غير منطقية وغير مقبولة. وتعتبر مصر من أكثر البلاد عددا فى السكان بالمنطقة، علاوة على أنها تعتبر من أكثر المجتمعات المتكيفة مع تكنولوجيا الحاسبات فى العالم العربى، بالإضافة إلى أن العنصر البشرى المدرب فى هذا المجال من أكثر صادراتها بالمنطقة. وقد أخذت الحكومة المصرية من جانبها عديداً من الخطوات الطموحة نحو الإصلاح الاقتصادى وتطوير التكنولوجيا. كما تم إتخاذ عديداً من الإجراءات مؤخرا لدعم صناعة البرمجيات فى مصر. وفى هذه الورقة البحثية يتم إلقاء الضوء على دور صناعة البرمجيات المصرية فى توفير فرص عمل ذات نوعية متميزة للشباب علاوة على منتجاتها من حزم البرامج المبتكرة والمتجددة وقدرتها على سد مطالب واحتياجات تكنولوجيا المعلومات للعالم العربى والخارجى. وهناك عديد من التحديات تواجه نمو هذه الصناعة تم التعرض لها ومناقشتها.

* النص الأصلى مكتوب باللغة الإنجليزية

الضبط الببليوجرافى الوطنى للبرمجيات فى مصر

د. مصطفى حسام الدين
أستاذ مساعد، قسم الوثائق والمكتبات
كلية الآداب، جامعة القاهرة

المستخلص

يصدر فى أغلب دول العالم التى تحرص على التعريف بإنتاجها من البرمجيات وتيسير الوصول إليه والتسويق له فضلا عن حمايته، ثلاث أدوات ببليوجرافية، بغض النظر عن شكل الإتاحة الذى تتوافر فيه هذه الأدوات:

الأداة الأولى: تحتوى عن كل برنامج مجموعة من البيانات المعيارية المقننة (تسجيلة ببليوجرافية) التى تصفه وتحقق جمهوره المقصود... إلخ، وتقدم مداخل مختلفة للبحث عنه منها الموضوع أو جانب التطبيق أو التشغيل الذى يعالجه، والشركة المنتجة، وتاريخ الإنتاج، والعتاد المستخدم، وبرامج التشغيل الملائمة.. إلخ.

الأداة الثانية: تهتم بتقديم ليس فقط مجموعة البيانات المعيارية المقننة عن كل برنامج، ولكن أيضا تضيف إليها نصوصا تتضمن تعليقات مفصلة عن خصائصه ومميزاته التى ينفرد بها، سواء من خلال ما يطرحه منتج البرنامج أو من خلال آراء الفنيين أو المتخصصين الذين خبروه أو طبقوه. وتقدم هذه الأداة أيضا مداخل الاسترجاع المتعددة عن كل برنامج.

الأداة الثالثة: تهتم بحصر وتقديم بيانات ببليوجرافية معيارية مقننة لكل ما كتب من أخبار أو مقالات أو تقارير أو حتى كتب عن برنامج معين، بهدف إتاحة الفرصة للباحث المدقق الذى يهتم بدراسة برنامج معين من الوصول إلى ماسبق أن كتب عنه.

ونظرا لأن أيا من تلك الأدوات المنهجية التي تحصر ما أبدعه المنتجون المصريون من برمجيات حتى الآن، وتعرف بها وتيسر الوصول إليها، باستثناء ما ينشر في بعض المجلات من بيانات غير كاملة أو إعلانات، أو ما يمكن تعرفه من خلال المعارض، فإن هذه الورقة تعرض لتصور عن كيفية إحكام الضبط الببليوجرافي الوطني للبرمجيات في مصر: الأولويات التي ينبغي الاهتمام بها، والجهة أو الجهات المعنية التي ينبغي أن تضطلع بهذه المهمة، والبيانات المعيارية المقننة التي ينبغي توافرها، وأساليب جمع البيانات، ومداخل الاسترجاع والبحث التي ينبغي إتاحتها، وطرق أو أشكال إصدار هذه الأدوات بانتظام واستمرار.

مفهوم الضبط الببليوجرافي الوطني ومتطلباته

★ المفهوم:

يعني به الجهود التي تبذل على المستوى الوطني لتحقيق السيطرة على أوعية المعلومات في دولة من الدول أو أمة من الأمم، وتتمثل هذه السيطرة في «تحقيق ذاتية مواد المعلومات المسجلة في كل وسائل الاتصال، وتقديم إمكانات الوصول المنطقية لكل مادة منها، وتقنين تجهيز وتداول البيانات المتعلقة بها». ويتضمن ذلك بطبيعة الحال إنشاء و / أو توفير التسجيلات التي تحتوي على البيانات الشاملة والمقننة والدقيقة التي تصف أوعية المعلومات في الدولة أو الأمة، وتمكن من الوصول إليها منطقيا من خلال مداخل استرجاع أو نقاط إتاحة متعددة، وماديا من خلال بيانات عن أماكن وجودها في مرافق المعلومات في هذه الدولة أو الأمة، كما يشتمل ذلك على جمع وتنظيم وبث هذه التسجيلات ضمن قنوات أو أدوات ببليوجرافية متعددة.

★ الأدوات:

● **أهمية الأدوات:**

تحقق هذه الأدوات الفائدة لكل المعنيين بهذه البرمجيات في البيئة العربية لغة وثقافة، تخطيطاً وإعداداً ونشراً وتطبيقاً واستخداماً:

● المعنيون بتخطيط وإعداد البرمجيات.

- الناشرون أو المنتجون والوسطاء.
- المستفيدون.
- المعنيون بصناعة العتاد (التجهيزات المادية)

- **أنواع الأدوات :**

- الببليوجرافيات الوطنية الجارية والراجعة، أو أدلة الحصر العامة
- أدلة عروض البرمجيات
- ببليوجرافيات تحصر ما كتب عن البرمجيات

★ المصادر

المباشرة :

- الإيداع تطوعاً
- الإيداع تنفيذا للقانون
- الإيداع تنفيذا مع الربط بحقوق الملكية الفكرية

غير المباشرة:

- المتابعة النشطة.

★ الهيئة المعنية

الهيئة الببليوجرافية الوطنية أو المكتبة الوطنية، وذلك بحكم:

- أهدافها.
- وظائفها.
- إمكاناتها أو ما ينبغى أن يتوافر لها من قدرات.

المحاولة المنهجية الأولى لحصر البرمجيات في مصر ومؤشراتها

* نشرت أول محاولة لحصر هذه البرمجيات فى العدد الرابع من مجلة «الاتجاهات الحديثة والمعلومات» يوليو ١٩٩٥.

* أبعاد الحصر وحدوده

* المؤشرات:

من خلال تحليل البيانات الببليوجرافية التى وردت فى هذه الببليوجرافية أمكن الوصول إلى بعض المؤشرات التى تلقى الضوء على السمات والخصائص البنيانية التى تميز الإنتاج المصرى من البرمجيات. وتتناول هذه المؤشرات:

- العدد ونسبة النمو
- نوعيات البرمجيات
- الجمهور المقصود
- السعر
- التوافقية
- العدد ونسبة النمو

جدول (١) : التوزيع الزمنى ونسبة النمو فى إنتاج البرمجيات

السنة	العدد	٪	نسبة النمو
١٩٨٦	١	٠,٥١	
١٩٨٧	ــ	ــ	
١٩٨٨	٤	٢,٠٥	
١٩٨٩	١٢	٦,١٥	
١٩٨ـ	٢	١,٠٢	١٩٩٠ الأساس (٩,٧٣٪)
١٩٩٠	٢٤	١٢,٣	١٢٦,٣٪
١٩٩١	١٨	٩,٢٣	٩٤,٧٪
١٩٩٢	٥٤	٢٧,٦٩	٢٨٤,٢٪
١٩٩٣	٦٣	٣٢,٣	٣٣١,٥٪
١٩٩٤	١٥	٧,٦٩	مجمل التسعينات ٩٢٦,٣٪
١٩٩ـ	٢	١,٠٢	(٩٠,٢٣٪)
المجموع	١٩٥	٩٩,٩٦	

● المنتجون

جدول (٢) : عدد منتجى البرمجيات ونسبة إسهاماتهم

م	المنتجـــــون	عـدد	٪
١	١ ـ منظمات حكومية دولية	١	٠,٥١
	٢ ـ شركات قطاع خاص		
٢	* خليفة للهندسة والكمبيوتر	٩٠	٤٦,١٥
٣	* صيخر (العالمية للبرامج	٤٣	٢٦,٥
٤	* شركة الخليج العالمية للتكنولوجيا المصرية	١١	٥,٦٤
٥	* انفو آراب (أنظمة المعلومات العربية)	٧	٣,٥٨
٦	* التكنولوجيا التبادلية المتقدمة	٦	٣,٠٧
٧	* الرواد للنظم والحاسبات	٥	٢,٥٦
٨	* دلتا كمبيوتر	٤	٢,٠٥
٩	* ايجبت سوفت	٤	٢,٠٥
١٠	* الشركة الاستشارية لهندسة الحاسوب	٣	١,٥٣
١١	* المركز العربى للحاسب الآلى (باك إيجبت)	٣	١,٥٣
١٢	* شركة الكمبيوتر المتقدمة والخدمات التعليمية	٢	١,٠٢
١٣	* العربية للأنظمة العلمية والهندسة	٢	١,٠٢
١٤	* مايكروتك	١	٠,٥١
١٥	* كمبيوتك	١	٠,٥١
١٦	* فانس سوفت	١	٠,٥١
١٧	* الشركة الشرقية للحاسبات	١	٠,٥١
١٨	* دايت سوفت	١	٠,٥١
١٩	* ديجيتال سوفت	١	٠,٥١
٢٠	* دوفت	١	٠,٥١
٢١	* الحاسبات المصرية	١	٠,٥١
٢٢	* باى للحلول العلمية	١	٠,٥١
٢٣	* ألفا مصر	١	٠,٥١
٢٤	* إيجبشيان داتا سيستم	١	٠,٥١
٢٥	* تيم العالمية	١	٠,٥١
٢٦	* الجيزة للنظم	١	٠,٥١
٢٧	* تكنولوجيا الأنظمة المصغرة التطبيقية	١	٠,٥١
	المجمـــــوع	١٩٥	٩٩,٩٣

● نوعيات البرمجيات وموضوعاتها

جدول (٣) : التوزيع النوعى الموضوعى للبرمجيات

نوعيات البرامج	العدد	٪
١ ـ برامج النظام System software		٩,٧٤
١ ـ ١ التعريب (لنظم التشغيل)	٧	
١ ـ ٢ الاتصـــالات	٦	
١ ـ ٣ دعم النظــام	٣	
١ ـ ٤ تطوير النظام	٣	
٢ ـ برامج التطبيقات Application software		
٢ ـ ١ عامة الأغراض		٨,٧١
* الوسائط المتعددة	١٠	
* معالجة الكلمات	٤	
* الجداول	٢	
* الرسوم البيانية	١	
٢ ـ ٢ إدارة الأعمال والمؤسسات (عامة)		١١,٢٨
* المحاسبة	٨	
* ميكنة المكاتب	٧	
* إدارة العمليات / مراقبة المخزون	٤	
* المبيعات	١	
* الأفراد / الأجور / تحليل العمالة ... إلخ	١	
* التسويق / تحليل المبيعات	١	
٢ ـ ٣ إدارة الأعمال والمؤسسات (نوعية)		٤,١
* المدارس	٣	
* المكتبات	٢	
* المقاولات	٢	
* العيادات	١	
٢ ـ ٤ لغوية		١٣,٨٤
* الخطوط	٢٣	

تابع جدول (٢) : التوزيع النوعى الموضوعى للبرمجيات .

نوعيات البرامج	العدد	%
* التدقيق النحوى والإملائى	٣	
* قواميس	١	
٢ ـ ٥ علمية		٢,٥٦
* تصميم هندسى	٢	
* تفاعل كيميائى	٢	
* تحليل معمارى	١	
٢ ـ ٦ إعلانات ورسوم فنية	٦	٣,٠٧
٢ ـ ٧ الطباعة والنشر	٩	٤,٦١
٢ ـ ٨ التعليم		٢٩,٢٣
* دروس في :		
ـ الحاسب الآلي	٢٠	
ـ الرياضيات	١٢	
ـ الدين	١٠	
ـ اللغات	٩	
ـ الجغرافيا	٣	
ـ السلوكيات	٢	
* تصميم الدروس والبرامج التعليمية	١	
٢ ـ ٩ التسلية والترقية		٩,٢٣
* اختبارات وألغاز	١١	
* قصص	٧	
٢ ـ ١٠ قواعد بيانات		٣,٥٨
* إدارة قواعد البيانات	٢	
* تكشيف نصوص	٣	
* ببليوجرافية	١	
* أشخاص	١	
المجمـــــوع	١٩٥	٩٩,٩٥

● الجمهور المقصود

جدول(٤) : توزيع البرامج وفقا للجمهور المقصود

الجمهور المقصود	العدد	٪
١ ـ المهنيون		٤٦,١٥
١ ـ ١ الطابعون والناشرون	٣٩	٢٠,٠٠
١ ـ ٢ أخصائيو الحاسب الآلي	٢١	١٠,٧٦
١ ـ ٣ أخصائيو المعلومات (تصميم قواعد البيانات وعرض وتحليل البيانات	١٨	٩,٢٣
١ ـ ٤ مصممو الإعلانات	٦	٣,٠٧
١ ـ ٥ المهندسون	٣	١,٥٣
١ ـ ٦ الصيانة	٢	١,٠٢
١ ـ ٧ المدرسون	١	٠,٥١
٢ ـ الأطفال والناشئة (برامج تعليمية وقصص)	٤٦	٢٣,٥٨
٣ ـ المديرون (الإدارة الإشرافية والوسطى والعليا)	٣٠	١٥,٣٨
٤ ـ الجمهور العام (التسلية وتعلم مهارات استخدام الحاسب الآلي	٢٩	١٤,٨٧
المجمـــــوع	١٩٥	٩٩,٩٨

● السعر

جدول (٥) : توزيع البرامج إلى فئات وفقاً لأسعارها

الجمهور المقصور	العدد	٪
مجانى	١	٠,٥١
أقل من ٥٠ جم	٥٨	٢٩,٧٤
من ٥٠ إلى أقل من ١٥٠ جم	١٥	٧,٦٩
من ١٥٠ إلى أقل من ٢٥٠	٨	٤,١
من ٢٥٠ إلى أقل من ٣٥٠	١٠	٥,١٢
من ٣٥٠ إلى أقل من ٤٥٠	٣	١,٥٣
من ٤٥٠ إلى أقل من ٥٥٠	٢	١,٠٢
٥٥٠ فأكثر	٤	٢,٠٥
غير معروف	٩٤	٤٨,٢
المجمـــــوع	١٩٥	٩٩,٩٦

● التوافقية

جدول (٦) : توزيع البرامج وفقاً للتوافقية

الفئة	العدد	٪	ملاحظـــــات
IBM والأجهزة المتوافقة معها	١٨٧	٩٥,٨٩	منها «١٦٥» برنامجاً تتطلب برمجيات نظام التشغيل DOS بنسبة ٨٤,١٣ و ٢٢ برنامجاً يتطلب برمجيات التشغيل WINDOWS بنسبة ١١,٧٦
MAC والأجهزة المتوافقة معها	٨	٤,١	
المجمـــوع	١٩٥	٩٩,٩٩	

تصور لمستقبل الضبط الببليوجرافى الوطنى للبرمجيات فى مصر

★ الأولويات

- الحصر الأمامى أولا
- الحصر الخلفى ثانيا

★ الهيئة المعنية بالحصر

- ضرورة توافر الإمكانات المادية والتجهيزية والبشرية والمالية.
- الهيئات المرشحة
  - الهيئة الببلوجرافية الوطنية ـ المكتبة الوطنية
  - الجمعيات العلمية المعنية
  - الجهاز المركزى للتعبئة العامة والإحصاء ـ المركز القومى للحاسبات

★ نظام حماية حقوق الملكية الفكرية

- القانون ٢٨ لسنة ١٩٩٢
- تطبيق الاتفاقيات الدولية

★ نظام للترقيم الدولى للبرمجيات

- تطبيق الترقيم الدولى للكتب (ISBN) على البرمجيات
- ترقيمة الإيداع

★ البيانات المعيارية

- قواعد الوصف المعيارية ISBN (ER)
- قواعد الفهرسة الانجلو أمريكية (AACR2)

★ **مداخل الاسترجاع**

- العنوان
- المنتج / العنوان
- التطبيقات / العنوان
- النظام / التطبيقات / العنوان

★ **الأدوات المقترحة**

- **قاعدة بيانات ببليوجرافية للبرمجيات**
- **مخرجات القاعدة**
  - تسجيلات
  - عروض البرمجيات
  - ببليوجرافيات عما كتب عن البرمجيات
- **أشكال الإتاحة**
  - مطبوع
  - مقروء آلياً
  - على الخط المباشر
  - على الإنترنت

الجزء الثانى

هندسة تطوير البرمجيات
النماذج والأدوات والأساليب

نمذجة مجردة لنظم المعلومات
الخطوة الأولى لميكنة مرحلة تحليل النظم *

د. محمد بدر السنوسى

المستخلص

ازدادت صعوبة إنشاء البرمجيات بدرجة كبيرة، حيث أدت الاحتياجات التقنية والمتزايدة للمستثمرين إلى الحاجة إلى أعداد كبيرة من مصممى ومبرمجى النظم ممن تتوفر لديهم القدرة على تحقيق درجة عالية من التعاون والتنظيم فيما بينهم، ويملكون الخبرة العملية لإدارة مشروعات تصميم وتطوير كهذه. وعلى الرغم من الوسائل المتوفرة لتطوير البرمجيات؛ خاصة فى مراحل التصميم (توجيه الهدف، وسائل التصميم عالية التخصص مثل NEWERA مع INFORMIX وPOWER BULIDER مع SYBASE و2000 DESIGNER مع ORACLE) فإن تعرض الأنظمة الممكنة للأعطال وتكلفتها العالية أصبحت أخباراً منتظمة فى كل مكان، مما يطلق عليه أزمة البرمجيات.

إن استخدام طرق وتقنيات جديدة تؤمن تحسين جودة الأدوات المستخدمة؛ خاصة فى مرحلة تحليل النظم سيساعد فى القضاء على هذه الأزمة فى المستقبل.

هذا البحث يهدف إلى البدء فى وضع إطار نموذج لمنظومة المعلومات وإعطاء تعريفات دقيقة لكل خطوات وكيانات عملية تحليل النظم. ولهذا فإن هذا البحث يبدأ بتوضيح معنى «مفهوم المنظومة ثم نقل هذا المفهوم إلى نموذج» من وجهة نظر المنظومة. كذلك سيوضح هذا البحث أدواراً دقيقة للمشاركين فى عملية التحليل. سيقدم هذا البحث نظرة مطورة للطرق والمنهجيات والأدوات المستخدمة فى ميكنة الخطوة الأولى لتحليل النظم، ومنها سنقدم تعريفاً للنموذج المقترح.

* النص الأصلى مكتوب باللغة الإنجليزية

تحسين معولية برامج الحاسب من خلال تطويرها بدمج أسلوب الغرفة النظيفة بالأساليب الشيئية *

د.علاء الدين محمد فهمى

المستخلص

يعد أسلوب هندسة البرامج باستخدام أسلوب الغرفة النظيفة هو أسلوب لتطوير البرامج بما يحقق تحسين جودتها وتقليل تكلفتها. وهذا الأسلوب أكتسب اسمه من الغرف النظيفة التى تستخدم فى إنتاج الشرائح الإلكترونية، حيث تركز أساليب التحكم الإحصائى فى الجودة على تلافى العيوب والأخطاء علاوة على إزالتها. ويركز أسلوب الغرفة النظيفة على معولية البرامج، ولكنه لا يتعرض لشرح تقنيات التصميم. وتقدم الأساليب الشيئية تقنيات خاصة بالتصميم والمعمارية تحقق القوة والترابط وإمكانية إعادة استخدامها وسهولة تعديلها. ويحقق دمج هذين الأسلوبين الحصول على أسلوب أكثر قوة وفعالية من كل أسلوب على حدة.

ويوفر هذا الدمج المعولية المطلوبة من خلال تحقيق ضبط البرامج، اعتماداً على نموذج الوظائف لأسلوب الغرفة النظيفة، علاوة على التصميم الموجه طبقاً لما هو متبع فى تطوير البرامج بالأسلوب الشيئى. وتوضح هذه الورقة البحثية كيف يمكن تحقيق هذا الدمج من خلال التحليل الشيئى مع مواصفات الصندوق الأسود بالغرفة النظيفة، مع اقتراح الوثائق اللازمة لدعم التحقق من التصميم الشيئى، كما تم عرض دليل لكيفية عمل التوازن المطلوب من خلال عملية الدمج.

---

* النص الأصلى مكتوب باللغة الإنجليزية

نحو تحسين برمجيات النظم الموزعة والمتوازية *

د. أحمد طه شهاب الدين، د. محمد منير عيسى

المستخلص

تكنولوجيا المعلومات تتقدم بمعدل هائل، ومع ذلك فإن معظم البرمجيات تواجه بعض المشاكل.

وبسبب التعقيد الشديد في النظم الموزعة أو المتوازية، فإن تطوير برمجياتها يتأثر بعوامل كثيره ويكون عرضه للأخطاء وأكثر تكلفة.

لكى ننتج برمجيات ذات كفاءة وجدوى اقتصادية للنظم المتوازية والموزعة لابد من بذل مجهودات فى جميع مراحل العملية الإنتاجية؛ خاصة المراحل الأولى منها وهى مرحلتى التوصيف والتصميم.

إن هندسة البرمجيات وأساليب تأكيد الجودة هما الأساس فى إنتاج برمجيات النظم الموزعة والمتوازية.

هذا البحث يقدم إطاراً يجمع بين أساليب هندسة البرمجيات وأساليب تأكيد الجودة يساهم فى إنتاج برمجيات النظم المتوازية أو الموزعة ذات الكفاءة العالية والجدوى الاقتصادية.

* النص الأصلى مكتوب باللغة الإنجليزية

التضمينات التكنولوجية فى تطوير البرمجيات *

أ. د. محمد محمد الهادى

المستخلص

تتسم صناعة الحاسب الآلى المنتشرة فى أرجاء العالم منذ بدايتها بالابتكار والتحسين المستمر، بالإضافة إلى التغير السريع المتلاحق. وقد نتج عن عمليات التطوير بزوغ أهمية البرمجيات المتنامية، ويتمثل ذلك فى البرامج التى تشغل الحاسبات الآلية، وتسمح لها بالاتصال بعضها مع بعض من خلال شبكات البيانات.

وتحاول مصر كدولة نامية الحصول على حصة فى صناعة المعلومات الدولية الحالية، كما تهدف الارتقاء بتطوير صناعة البرمجيات الخاصة بها من خلال تنوع السياسات ومقاييس الصناعة. وفى الوقت نفسه، بدأ عدد متنامٍ من شركات ومنظمات تطوير البرمجيات فى تطوير البرامج لكى تسوق داخليا وخارجيا.

وتعتبر صناعة البرمجيات الدولية صناعة عقلية إلى حد كبير. وأى تقدير وتقويم لصناعة وسوق البرمجيات على المستوى الدولى قد يعتمد على بيانات غير كافية ومتضاربة وغير موثوق منها. ويجعل ذلك من التنبؤات لمستقبل منتجات وخدمات البرمجيات عملية تحمل كثيرا من المخاطرة إلى حد كبير. وعلى الرغم من ذلك فقد عملت هذه التقديرات والمدى المستخلص أنه من ٧٠ إلى ١٨٠ بليون دولار أمريكى تمثل الناتج من هذه الصناعة فى بداية التسعينيات، الذى تضاعف فى الوقت الحاضر.

وتتواجد أربع قوى مؤثرة على الارتقاء بالتغييرات السريعة لصناعة البرمجيات فى العالم حاليا. وتتمثل هذه القوى الأربع فيما يلى:

(١) الإنتاجية المتدنية فى البرمجة مع البرمجيات والأنشطة المساندة المرتبطة بها تمثل حاليا التكاليف المتعاظمة من تكاليف النظام الكلى.

* النص الأصلى مكتوب باللغة الإنجليزية

(٢) الابتعاد من حلول البائع الفردى كطريقة ملائمة للمنظمات لتلبية حاجاتها من نظم المعلومات إلى الحلول المعتمدة على تفصيل الأجهزة والبرمجيات من قبل البائعين المتكاملين والمتعددين.

(٣) زيادة التأكيد على إنتاج البرمجيات ومبيعاتها بواسطة بائعى وموردى الأجهزة، أدت إلى زيادة إنشاء المنشآت المتوسطة والكبيرة.

(٤) أدى توسع الصناعة وتشتتها إلى وجود عدد كبير من بائعى البرمجيات المستقل كل منهم عن الآخر.

وقد أصبحت تكنولوجيا تطوير البرمجيات المرتبطة بالأدوات والطرق والنماذج التنظيمية والمعرفة التكنولوجية المستخدمة فى تحويل المدخلات إلى مخرجات، عنصراً متزايد الأهمية فى التنافس الدولى للمنظمات، وإدارة عملية انتاج البرمجيات فى المنشآت المنتجة.

وتستعرض هذه الورقة الفنية بعض مزاولات تطوير البرمجيات، التى أدخلت من قبل منشآت تطوير البرمجيات للمساعدة فى ترشيد تكاليف هندسة البرمجيات والرقابة عليها وتحسين كل من المنتجات النهائية وعملية تطويرها. وقد عملت الدراسة المقدمة على فحص بعض التكنولوجيات الحديثة التى تؤثر على عملية انتاج البرمجيات، وتتطلب إعادة توجه فى التفكير عن استثمارات الخيارات التكنولوجية.

ومن هذا المنطلق، تلقى هذه الورقة الفنية الضوء على نموذج نضوج القدرة CMM، الذى يوفر لمنشآت ومنظمات تطوير البرمجيات إطارا مناسبا لتقويم عمليات تطويرها والقيام بخطوات ملائمة لتحسينها. وتتضمن الخمس طبقات لهذا النموذج: الطبقة التمهيدية حيث تعمل الشركة أو العمل على تطبيق البرنامج؛ الطبقة الثانية تتمثل فى قدرة منشآت تطوير البرمجيات على إمداد منتجاتها من البرامج على أساس جدول متناسق وبمستوى جودة معينة. وفى المستوى أو الطبقة المفسرة لعمليات التطوير، يبدأ المطورين فى فهم المزاولات والإجراءات التى تدعوهم إلى إمداد البرامج بطريقة متناسقة، وتعمل على توزيع الإجراءات خلال منشآتهم؛ وعلى المستوى الإدارى للنموذج، فيرتبط بالقياس الكيفى للإجراءات والمزاولات المتنوعة؛ أما فى مستوى المثالية تنشأ البرامج الجديدة كروتينات كما تحتفظ بالتحسينات الدائمة.

الجزء الثالث

معايير ومواصفات تطوير البرمجيات

نحو أسلوب لتقييم جودة برامج الحاسب *

د. علاء الدين محمد فهمى

المستخلص

من أجل مواجهة قضية جودة برامج الحاسب، والتى بدأت تزداد أهمية مع الانتشار الكبير فى استخدامها فى شتى المجالات تم عمل الأيزو ٩١٦٢ . وهذا المعيار يحدد ست خصائص للجودة والتى روعى أن تكون شاملة. ولكى يمكن عملياً وبنجاح تطبيق الأيزو ٩١٦٢ فإن هذه الورقة البحثية تقترح إطاراً لتقييم البرامج على ضوء الخصائص الواردة فى هذا المعيار العالمى. وهذا الإطار تم وضعه بحيث يسمح بالتغييرات التكنولوجية دون الحاجة إلى تعديل أى من قواعده الرئيسية. ويحتوى الإطار المقترح على أربعة خطوات للتخطيط والتحكم وتقرير نتائج التقييم. وهذه الخطوات، هى: توصيف مطالب التقييم، تصميم التقييم، وإدارة عملية التقييم، وأخيرا إعداد التقارير الخاصة بالنتائج التى تم التوصل إليها أثناء التقييم.

* النص الأصلى مكتوب باللغة الإنجليزية

نموذج تحليلى لتقييم تأثير تكنولوجيا المعلومات على مستويات الإدارة *

د. نشأت الخميسى الغيطانى

المستخلص

لا أحد ينكر أهمية بل ضرورة تطبيق تكنولوجيا المعلومات لميكنة أنظمة العمل فى المجالات والتنظيمات المختلفة، على الرغم من اضطرار القائمين بمشروعات الميكنة للمواجهة العلمية المدروسة لشتى صور المقاومة المباشرة وغير المباشرة فى منظماتهم لإنجاح مخططات التطوير؛ مما يضفى أهمية كبيرة للبحوث التى تتناول التأثيرات المختلفة لنظم المعلومات المعتمدة على الحاسب الآلى على الهياكل التنظيمية وأساليب وتقنيات دورات العمل، وتأثير ذلك على مستويات الإدارة المختلفة. وعلى الرغم من نجاح عديد من الأبحاث فى نمذجة وتحليل وتصنيف مستويات وأنواع نظم المعلومات المختلفة، إلا أن هناك القليل من البحوث المنشورة، التى تتناول تحليل دقيق ونمذجة لدراسة وتقييم تفاعل المستويات الإدارية المختلفة، مع أنظمة عمل تعتمد على تكنولوجيا الحاسبات والبرامج.

ولقد استعرض خلال هذا البحث تصنيفات نظم المعلومات المختلفة، مع التركيز على الخصائص المميزة لكل منها من منظور وظائفها وتأثيراتها على أنظمة العمل المختلفة كتقديم لنموذج كمى تحليلى بهدف الوصول لبعض المؤشرات والنسب التحليلية لمدى تأثر كل من المستويات الوظيفية المختلفة بتطبيق تكنولوجيا المعلومات، فى محاولة لإجابة عديد من الأسئلة بشكل تطبيقى، ومنها على سبيل المثال:

* إلى أى حد يمكن مع الاستخدام الأمثل لتكنولوجيا المعلومات إعادة النظر فى هيكلة التنظيم والاستغناء عن عدد من كوادره البشرية؟

* النص الأصلى مكتوب باللغة الإنجليزية

* إلى أى نسبة يمكن الجزم بأن طاقة المديرين المهدرة أو غير الموجهة نتيجة لقصور النظام اليدوى يمكن أن تحرر لصالح التطوير فى ضوء تطبيق التكنولوجيا المعلوماتية؟

ولقد تم عمل دراسات تطبيقية لنمذجة متطلبات مستويات الإدارة المختلفة شملت أكثر من ١٥٠ مدير لكل من مستويات الإدارة فى عدد خمسة عشر شركة من قطاعات التشييد والاستطلاح مع الاسترشاد ببعض النماذج المقارنة؛ حيث تم من خلالها تنقيح النموذج المقترح بشكل أقرب للواقع التطبيقى وبتطبيق النموذج المقترح ومعايير القياس التحليل الواردة به، تضمن البحث بعض النتائج التطبيقية على العينة المشار إليها منها على سبيل المثال لإجابة بعض الأسئلة المشار إليها أنه مع التطبيق الناجح. ومع المراحل النهائية لمشروعات الميكنة يمكن أن تتراوح نسبة التفاعل أو الاستغناء عن بعض الأعداد أو من منظور آخر تعظيم القيمة المضافة لطاقة المديرين بنسبة، تتراوح من ٣٠٪ - ٧٠٪ طبقاً لمستوى الإدارة سواء أكانت إشرافية أم وسطى أم عليا، إضافة إلى توضيح بعض التفاعلات على دورات وأساليب العمل المختلفة.

تكنولوجيات شبكات الوسائط المتعددة *

د. محمد منير عيسى، د. علاء الدين محمد الغزالي

المستخلص

إن أنظمة المؤتمرات المكتبية والتطبيقات الأخرى الموزعة تتطلب شبكات رقمية لنقل البيانات السمعية والمرئية (الفيديو). إن عديداً من الشبكات الحالية غير مناسبة لتلك التطبيقات، ذلك أن البروتوكولات التي تحكم هذه الشبكات الحالية غير مناسبة، وكذلك النطاق المحدود غير كافٍ لحركة سير السمعيات والمرئيات (الفيديو) الرقمية.

من ناحية أخرى فإن شبكات الوسائط المتعددة تمكن من ترابط وتوافق أفضل لخصائص (طبيعة أو دور) الاتصال لتطبيقات الوسائط المتعددة الموزعة.

هذه الشبكات بصفة خاصة صممت لحركة مرور الوسائط المتعددة، وتختلف شبكات الوسائط المتعددة عن الشبكات المحلية والواسعة الحالية فى طرق عدة (عديدة). إنها تتميز بميزات بارزة (سمات) مختلفة. هذه السمات (الميزات البارزة) قدمت ونوقشت من خلال هذا البحث.

---

* النص الأصلى مكتوب باللغة الإنجليزية

الجزء الرابع

برمجيات الوسائط المتعددة وبرامج التعليم الذاتى والنظم الخبيرة

برامج إثرائية للتعلم الذاتى فى مناهج المرحلة الابتدائية باستخدام الوسائط المتعددة

أ.د. عايدة عباس أبو غريب ، د. شعبان حامد على

المقدمة

يفرض عصر الانفجار المعرفى متطلبات جديدة تهدف تمكين تلاميذنا من استيعاب عناصر المعرفة ومهاراتها ووسائطها، وحسن استخدامها وتوظيفها، كما يستلزم منح التلاميذ قدراً أكبر من المسئولين فى اكتساب المعرفة والتعليم الذاتى.

ويعبر هذا الاتجاه عن منظومة إنتاجية، تسعى إلى استخدام وتطبيق أساليب التكنولوجيا الحديثة، وما تقتضيه من تشغيل منطقى للعمليات العقلية فى عمليات التعليم والتعلم، وإلى تطويع الأجهزة والمعدات ذات القدرات الفائقة فى عرض وتخزين وتحليل واستقصاء المعلومات للعملية التعليمية من خلال مواد وبرامج ذات أهداف سابقة التحديد. ومن ثم صار تصميم وبناء وتنفيذ برامج إثرائية للتعلم الذاتى فى المناهج الدراسية باستخدام الوسائط المتعددة، ضرورة تفرضها طبيعة العصر الذى نعيشه، والذى يعرف بعصر الثلاثة "C_S" The age of three "Cs" وهم عصر الحاسب الآلى وعصر الاتصال، وعصر التحكم، وشكل ذلك حاجة ملحة لإتمام الدراسة الحالية، والتى تأتى مكملة للمشروع القومى لإدخال الكمبيوتر فى التعليم، والذى يهدف تنمية قدرة الطلاب على التفكير والتكيف مع مقتضيات العصر والتطور العلمى والتكنولوجى.

الإطار النظرى

يعتبر اكتساب التلاميذ مهارات التعلم الذاتى من خلال ممارسة أنشطة إثرائية مصاحبة للمناهج الدراسية من خلال تصميم برامج تنفذ عن طريق الكمبيوتر، ويعد ذلك من أهم

الأهداف المعاصرة التى تسعى لتحقيق كافة النظم التعليمية، حيث إن التعلم الذاتى كما يرى مارشيز (Marchese , 1997) يتميز بأنه:

- يوفر قدرة متزايدة ليقوم مدى تقدمه ونجاحه.

- إفساح مجال أكبر للتلاميذ للمشاركة وتحمل المسئولية.

- يؤدى إلى تعلم ناجح للمادة الدراسية ويصل بها إلى مستوى الإتقان.

- ينمى اتجاهاً إيجابياً تجاه التلاميذ أنفسهم والمدرسة والمادة الدراسية.

ويتضمن التعلم الذاتى مكآفاته في ذاته، لأنه يحقق مجموعة من الدوافع الأساسية لدى التلاميذ والتى من أهمها الدافع المعرفى Cognitive drive، ويتمثل هذا الدافع فى الرغبة فى المعرفة والفهم وإتقان المعلومات وصياغة المشكلات وحلها، وهو مشتق بطريقة عامة من دوافع الاستطلاع Curiosity، والاستكشاف Exploration والمعالجة Manipulation، ويرى أوزوبل Ausubel, (Ausubel, D.P., 1968) أن تلك الدوافع لها خصائص دافعية احتمالية غير محددة في محتواها، وتتحقق قوتها الدافعية في التعبير والممارسة، وتتحدد فى الاتجاه مع نمو التلميذ معرفياً نتيجة الممارسة الناجحة، وتوقع النواتج المشجعة من الممارسة التى قد تُؤدى فى المستقبل.

ويؤكد فكرى ريان (١٩٨٩) أهمية ممارسة التلاميذ للتعلم الذاتى حيث إن تفاعلهم مع الخبرات الهادفة المباشرة والمشكلات الحية تنمى لديهم مفهوماً إيجابيًا للذات، وقبول الآخرين وتحسن من أدائهم للاشياء، وأن ما يحصلونه من معارف نتاج تفاعلهم مع المواقف التعليمية.

وباستقراء الأدبيات التربوية والنفسية يلاحظ أن النظريات النفسية أظهرت أهمية الأنشطة الإثرائية، فيؤكد ميلر (Miller, 1974) أن نظرية بياجيه Biagete أعطت وظيفة بيولوجية واضحة للأنشطة الإثرائية، باعتبارها تجارب ومواقف ومشكلات تعمل، تمثل وتهضم Assimilation المواقف التعليمية، وتقدم وصفاً متماسكاً لنمو الأنشطة المتتابعة لكل مرحلة عمرية يمر بها التلميذ.

ويرى برونر Bruner أن الجهد الذى يبذله التلميذ خلال اكتشافه لخبرات التعلم الذاتى تجعله يمارس أنواعاً من الاستدلال والاستبصارات والتعميمات فى سبيل الوصول للمعرفة.

وقد اتفق علماء التربية منذ جان جاك روسو، وجون ديوى وحتى المعاصرين منهم على ضرورة ممارسة الأنشطة الإثرائية من جانب المتعلمين، وأنه لا تعلم ذو معنى دونها، حيث تجعل التلميذ مفكراً ومجرباً وإيجابياً، وتنمى لديه جوانب مهمة فى شخصيته مثل القدرة على اتخاذ القرار والاستقلالية والاعتماد على الذات وتنمية قدرات الإبداع.

ويمكن إجمال وظائف البرامج الإثرائية المصاحبة للمناهج فى الآتى:

١- وظيفة سيكولوجية وتربوية:

تحقق ممارسة التلاميذ لأنشطة مصاحبة للمناهج الدراسية مجموعة من الوظائف النفسية من أهمها تنمية الميول والمواهب وقضاء أوقات الفراغ فى نشاط مثمر، وهو ما يساعد فى تحقيق الصحة النفسية، وأحد السبل المهمة للتوجيه الدراسى والمهنى، كما توفر الدافعية للتعليم داخل الفصل وتساعد على رفع مستوى الإنجاز.

وتؤكد ذلك عديد من الدراسات، منها دراسة تجريبية أجراها كل من ولتر Walter L.B، وجارى Gary D.Y (١٩٨٥) بهدف تعرف أثر وجود عينات حية من الكائنات داخل الفصل الدراسى، يمارس التلاميذ عليها أنشطة كشفية وعمليات استدلالية مصاحبة للمنهج، وقسمت عينة الدراسة إلى مجموعة تجريبية ومجموعة ضابطة، وتوصلت الدراسة إلى أن تلاميذ المجموعة التجريبية والتى مارست الأنشطة المصاحبة قد حققوا مستوى تحصيليًّا مرتفعاً، واكتسبوا اتجاهات إيجابية نحو المنهج الدراسى، خلال زمن التجربة التى استمرت ١٦ أسبوعاً.

وفى دراسة أجراها نيوتن Newton, C.C (١٩٩١) بهدف تعرف أثر المشاركة فى أنشطة متنوعة من منهج إثرائى Extra curricular، ونوادى نشاط مختلفة على خصائص طلاب المرحلة الثانوية، وشملت عينة الدراسة (٤٢٥) طالب وطالبة، وتوصلت الدراسة إلى أنه يوجد ارتباط دال إحصائياً موجب بين عدد الساعات التى يقضيها الطلاب فى ممارسة الأنشطة، وبين تحسن خصائصهم النفسية والاجتماعية ونمو مفهوم الذات لديهم.

وفى دراسة تقويمية أعدها فين Fine، وفريدمان Fridman (١٩٩١). يهدف تقويم فعالية تدريس العلوم والجغرافيا بالمرحلة الابتدائية باستخدام الكمبيوتر، وشبكات المعلومات، وتكونت عينة الدراسة من ٥٦ مدرساً و ١٨٠٠ تلميذ فى ولاية أيوا Iowa بالولايات المتحدة

الأمريكية، توصلت الدراسة إلى أن المشروع يتسم بالفعالية فى تنمية التفكير الناقد والقدرة على حل المشكلات لدى التلاميذ، مما حقق رضا المعلمين عن استخدام التكنولوجيا الحديثة فى تدريس العلوم والجغرافيا، وفى دراسة أجراها كل من محمد أمين حسن، ورؤوف عزمى توفيق (١٩٩٧) استهدفت تعرف أثر برنامج يستخدم الوسائط المتعددة فى تنمية مفاهيم التربية الوقائية لدى تلاميذ الصف الخامس الابتدائى، وتنمية التفكير الابتكارى وبناء أثر التعلم لديهم، واستخدمت الدراسة الأدوات الآتية:

- برنامج التربية الوقائية معد باستخدام الوسائط المتعددة بالكمبيوتر.
- اختبار تحصيلى فى مفاهيم التربية الوقائية.
- اختبار التفكير الناقد.

وتوصلت الدراسة إلى فعالية البرنامج فى التحصيل الدراسى وتنمية التفكير الابتكارى لدى التلاميذ عينة الدراسة، ومن ثم يوضح تقبلهم للبرنامج.

كما أظهرت بقاء تعلمهم للبرنامج، كما اتضح من نتائج التطبيق المرجأ للاختبار التحصيلى.

٢ ـ الوظيفة الاجتماعية:

تساعد ممارسة التلاميذ للأنشطة المنهجية والمصاحبة فى مجموعات وخلال جماعات على تحسين فرص الاتصال بينهم، وقيام أواصر المودة والصداقة بينهم، وتمنحهم فرص ممارسة الديمقراطية وتحمل المسئولية والتعاون والثقة بالنفس وبالغير واحترام الأنظمة، والتوفيق بين صالح الفرد والجماعة، وتقدير القيمة العالية لأوقات الفراغ فى استثمارها.

وتحقق ممارسة الأنشطة الإنمائية مبدأ إيجابية التلميذ، وتعمق فعالية وأهداف المناهج الدراسية التى تدور حولها الأنشطة، وتكشف عن الميول المهنية لدى التلاميذ وتصقل مواهبهم العلمية.

والمدرسة كوكالة للتنشئة الاجتماعية تستنفر قيام التلميذ بممارسة الأنشطة الإثرائية المعارة بالكمبيوتر لتنمية ولاءاته وعواطفه، التى تتخطى حدود الأسرة إلى المجتمع المحيط به والوطن.

وقد تناولت دراسات عديدة الوظيفة الاجتماعية للأنشطة التعليمية، منها دراسة أجراها هورن (Horn Halot, 1978) بهدف تعرف أثر الأنشطة على عضوية الجماعة، واشتملت عينة الدراسة (٨٦) تلميذاً بإحدى المدارس الابتدائية الألمانية، وتوصلت الدراسة إلى أن فرص التعاون والمودة والصداقة يزيد أثناء ممارسة التلاميذ للأنشطة التعليمية.

وفى دراسة أجراها عبد الرؤوف محمد بدوى (١٩٨٨) بهدف تقويم الأنشطة التى تمارس فى المدرسة فى ضوء أهداف التربية الاجتماعية، وتوصلت الدراسة إلى أن النشاط المدرسى بصفة عامة مجالاً صالحاً لإشباع حاجات الطلاب وتهذيب ميولهم واكتشاف قدراتهم ورعايتها.

كما أكدت دراسة قام بها جوان (Joan, 1992)، وكذلك دراسة أجراها توم (Tom, 1995) أن استخدام الوسائط المتعددة وتوفير مكتبة تكنولوجية، تتكون من أسطوانات ليزر وكمبيوتر يساعد على تعلم المهارات وغرس القيم وتنمية الاتجاهات نحو التكنولوجيا

٣ـ الوظيفة التحصيلية:

لم يعد المنهج الدراسى مجرد المعارف والمهارات التى يتم تعليمها للتلاميذ، ولكنه صار يعنى جميع الخبرات وبرامج الأنشطة المصاحبة التى تثرى العملية التعليمية، والتى تمكن التلاميذ من نمو ذواتهم وشخصياتهم فى جوانبها المتعددة نمواً يتسق مع الأهداف التربوية، وقد قامت دراسات عديدة أكدت على الأثر الإيجابى للأنشطة الإثرائية والمنهجية على رفع المستويات التحصيلية لدى التلاميذ ومنها:

- دراسة أجراها كل من عمر عبد العزيز محمود، سعد عبد الرسول (١٩٩٢)، بهدف تعرف أثر النشاط ونوعيته على تحقيق التفوق الدراسى لعينة الدراسة التى اشتملت على تلاميذ ثلاث مدارس بإحدى مراكز محافظة أسوان، وتوصلت الدراسة إلى وجود علاقة دالة موجبة بين ممارسة التلاميذ لأنشطة مصاحبة للمناهج وتحصيلهم الدراسى، كما أن تنوع الأنشطة وتعددها لها أثر واضح فى مدى تحصيل الطلاب.

- دراسة أجراها رضا محمد السعيد (١٩٩٢) حول البرامج الإثرائية كرؤية مستقبلية لتطوير مناهج الرياضيات وممارسة الأنشطة الإثرائية، وضرورة توجيه انتباه الطلاب إلى الموضوعات الرياضية غير الموجودة بمقرارتهم الدراسية.

وفى ضوء التقدم العلمى والمعرفى الهائل فى العصر الحالى يستحيل الاكتفاء بالمناهج والكتب المدرسية للوصول إلى الكفاية النوعية للتلاميذ، نتيجة لعدم قدرة المناهج الدراسية على ملاحقة الجديد من المعلومات واكتفائها بتقديم أساسيات المعرفة وركائزها.

ويشير بوبرت (Pupert S.,1993) إلى أن استخدام الحاسب الآلى (الكمبيوتر) فى تنفيذ وعرض برامج إثرائية ليس بوصفه طريقة للتعلم الفردى أو الذاتى فقط، بل «آلة للمعرفة»، وهو تصور مستقبلى لكيفية استطاعة الوسائل التكنولوجية الجديدة تغيير علاقة التلاميذ بعملية التعلم بمن فيهم المبتدئين (الصفوف الأولى للمرحلة الابتدائية)، حيث تساعد التلميذ على التخلص من «الاعتماد على الحرف»، ويعنى به قدرة معرفة القراءة قبل التمكن من الحصول على أى نوع آخر من المعرفة، وبذلك يمكن توفير للتلاميذ وسيلة لصيانة طرقهم الخاصة «لقراءة العالم».

ويرى عبد العظيم الفرجانى (١٩٨٧) أن هناك كثيراً من التحديات يمكن سواجهتها باستخدام الكمبيوتر، مثل:

- التوسع الأفقى فى التعليم، وزيادة أعداد الدارسين مع النمو العددى للسكان والإقبال المتزايد على التعليم
- التدفق المعرفى وزيادة كمية ونوعية المعلومات المستمر.
- تعدد مصادر المعرفة وأوعيتها، فكيف يعتمد التلميذ على الكتاب المدرسى كمصدر وحيد للتعلم فى الوقت، الذى توجد فيه وسائط متعددة تثرى المنهج وتحقق أهدافه مثل الأفلام والشرائط السمعية والمصورة والمصغرات وبرامج الكمبيوتر.
- مشكلات التعليم الجمعى، ومشكلات الفروق الفردية.

وفى دراسة لكاجاز (Chagas, 1993) عن استخدام الفيديو التفاعلى، ودراسة سميث (Smith, 1994)، وأيضا دراسة جانجولى (Ganguly, 1995) عن التليفزيون التعليمى، توصلت كل منها إلى أن تحسين ظروف التعلم وتحسين مناخه وتحسين نمط التفاعل بين التلاميذ والمعلم يتم عن طريق تصميم البرامج التعليمية وتنفيذها باستخدام تكنولوجيا التعليم، بالإضافة لإزالة الرهبة لدى التلاميذ من التعامل مع المستحدثات التكنولوجية فى التعليم.

واستخدام الوسائط المتعددة فى تصميم وتنفيذ وتدريس برامج إثرائية يسمح بتوظيف المعلومات من خلال المميزات الآتية:

- تتناول أجزاء كبيرة من المعلومات.

- تراعى احتياجات التلاميذ.

- تسمح للتلميذ باكتساب المعلومات بالكمية والوقت المناسبين له.

- إثارة الفكر والخيال وتنمية التفكير الإبداعى.

- جعل المنهج مرناً بإدخال تعديلات على تنظيماته النمطية.

- بقاء أثر التعلم فى زيادة استخدام الحواس والتفاعل مع البرامج.

- علاج مشكلات التعلم التقليدى مثل: عدم الانتباه وعدم التركيز، والملل مما يزيد من فعالية التعليم.

- ترسيخ مبدأ التعلم الذاتى والاعتماد على النفس، واعتماد أثر التعلم.

وتتكون برامج الوسائط المتعددة Multi Media كمنهج دراسى مدار بالكمبيوتر، وليس مجرد معينات للتعلم من الآتى:

- المادة التعليمية (المحتوى)، والتى تقترح بحيث تحقق فلسفة وأهداف.

- أجهزة ووسائط تعليمية والتى تحول عن طريقها المادة التعليمية من الشكل اللفظى على المستوى الورقى (المكتوب) إلى مادة تعليمية حية، تستخدم فى تنفيذها الرسوم المتحركة والصور الحية والثابتة والأصوات... إلخ؛ كل ذلك مع إبداع المصمم والمنفذ والمخرج.

- أجهزة كمبيوتر بأدواته التعليمية (وحدات إدخال، لوحة مفاتيح، فارة، وحدة معالجة مركزية، شاشات، اسطوانات C.D.ROM مسجل عليها البرامج).

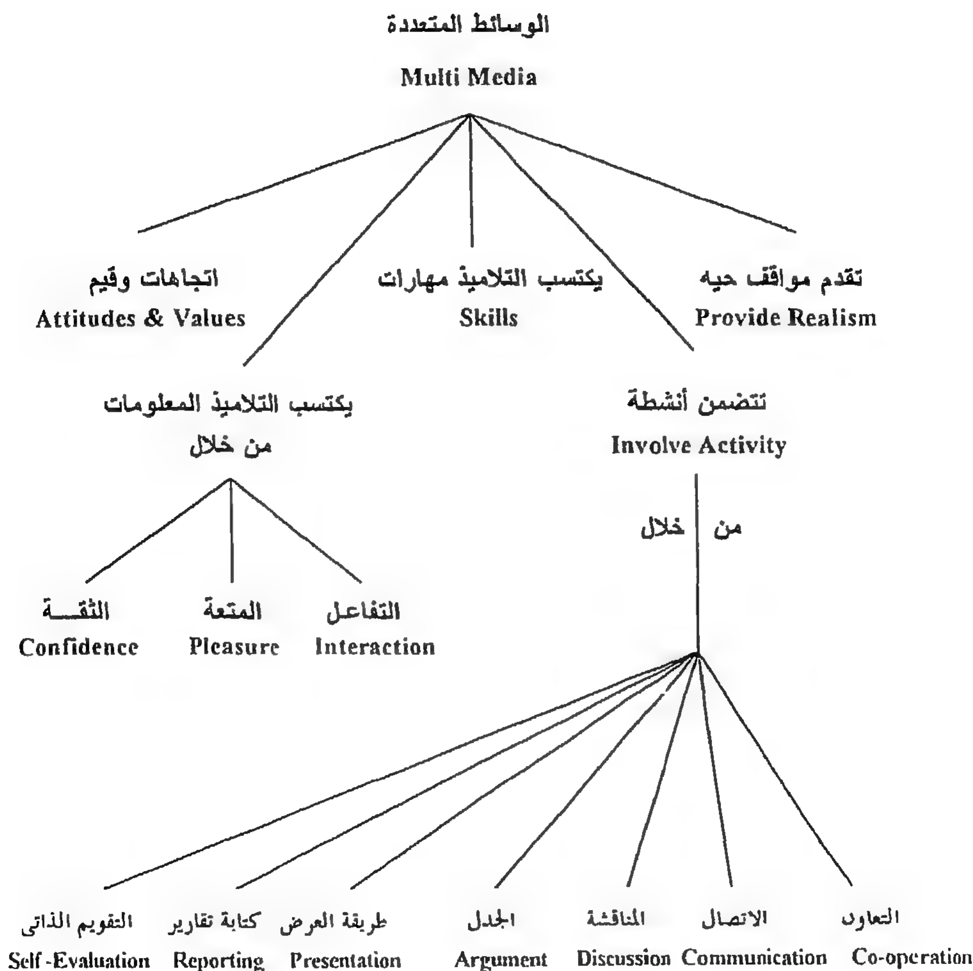
- طرق التفاعل وانتقال وتبادل المعلومات.

- الصياغة التى يتم عرض الموضوع بها (شاشات العرض).

- نظم التقويم والتدريبات على الأنشطة والأنتقال بين الشاشات.

- تغذية راجعة (معلومات أو تدريبات وأنشطة إضافية حول الموضوعات)، وذلك لتقديم المزيد للتلميذ إذا كان يحتاج إلى تدعيم ما تعلمه، أو إخباره بأنه أنجز المطلوب منه وتدعوه للتقدم فى البرنامج.

ونعرض أهم خصائص البرامج التعليمية المنفذة من خلال الوسائط فى التصور التالى :



ويتضح من العرض السابق مدى أهمية الأنشطة الإثرائية للمناهج الدراسية المصممة والمنفذة بالكمبيوتر وباستخدام الوسائط المتعددة، ويتضح أيضا ضرورة ممارسة التلاميذ لها كأنشطة مصاحبة أو موازية للمناهج، ويشير الواقع التربوى فى مدارسنا إلى وجود وضع غير متوازن بين أهمية تلك الأنشطة، وواقع وجودها فى المدرسة المصرية.

وتتفق النتيجة السابقة مع نتائج الدراسة التحليلية التى قام بها فايز مراد مينا (١٩٨٠)، والتى توصلت إلى أن الأنشطة التعليمية تمارس من جانب عدد قليل من التلاميذ وتتسم بالقصور، وأن الأسس الفكرية للنشاط المدرسى غير واضحة، وأن نظامنا التعليمى بوضعه الحالى يفتقد لهذا النوع من الأنشطة التربوية ولا يسمح بالأخذ بالمعايير المتصلة به.

مما سبق تتضح أهمية تناول قضية الأنشطة الإثرائية للمناهج الدراسية بالدراسة وإعداد نماذج لها، وشكلت حاجة ملحة لتصميم وبناء وتنفيذ برامج إثرائية للتعلم الذاتى مصاحبة للمناهج الدراسية باستخدام الوسائط المتعددة وتدريب المعلمين والموجهين ومديرى المدارس عليها، وكان طبيعياً أن تكون البداية مع مناهج المرحلة الابتدائية، حيث تعتبر الدراسة الحالية دراسة رائدة فى مصر، وتضافرت جهود باحثى شعبة بحوث تطوير المناهج بالمركز القومى للبحوث التربوية والتنمية كمنظرين ومصممين ومشرفين على التلاميذ والتطبيق لهذه البرامج مع إمكانات مركز التطوير التكنولوجى بوزارة التربية والتعليم؛ حيث تم الاتفاق بين الجهتين الفنية والتكنولوجية على تنفيذها، وذلك باشتراك مجموعة من معلمى المرحلة الابتدائية.

مشكلة الدراسة

تتناول الدراسة الحالية جانباً تربوياً مهماً يثرى نظامنا التعليمى ومناهجه الدراسية، وهو تصميم وبناء وتنفيذ برامج وتنفيذ برامج إثرائية لمناهج المرحلة الابتدائية باستخدام الوسائط المتعددة Malti Media، ويمكن طرح مشكلة الدراسة فى الأسئلة الآتية:

١ - ما معايير إعداد وتصميم برامج إثرائية لمناهج المرحلة الابتدائية باستخدام الوسائط المتعددة؟

٢ - ما التصور المقترح لبرامج أنشطة إثرائية مصاحبة لمناهج المرحلة الابتدائية، والذى

يمكن أن يمارسه التلميذ من خلال أساليب التعلم الذاتى بمعاونة الحاسب الآلى Computer Assisted Instruction (CAL)؟

ويتفرع من السؤال الثانى الأسئلة الفرعية الآتية :

أ - ما الأهداف التعليمية التى يسعى كل نشاط مقترح مصاحب للمناهج الدراسية (لغة عربية / لغة انجليزية / رياضيات / علوم / دراسات اجتماعية / أنشطة ومجالات) لتحقيقها ؟

ب - ما محتوى كل نشاط مقترح مصاحب للمناهج الدراسية (لغة عربية / لغة انجليزية / رياضيات / علوم / دراسات اجتماعية / أنشطة ومجالات) ؟

ج - ما أسلوب التنفيذ المناسب لكل نشاط مقترح مصاحب للمناهج الدراسية؟

د - ما أسلوب التقويم (تكوينى / تجميعى) المناسب لكل نشاط مقترح؟

٣ - ما فاعلية البرامج المقترحة كأنشطة إثرائية للمناهج الدراسية بالمرحلة الابتدائية؛ وفقاً لآراء لجان تحكيم بعد تنفيذها وقبل طرحها للميدان ؟

أهمية الدراسة

يتفق موضوع الدراسة الحالية مع الاتجاه العالمى الذى يعطى للأنشطة الإثرائية المصاحبة للمناهج الدراسية أهمية خاصة فى تنمية الجوانب الفكرية والسمات الشخصية لدى التلاميذ مثل تنمية قدرات التعلم الذاتى والاستقلالية وروح المبادأة والابتكار والإبداع، و نمو مفهوم للذات إيجابى وواقعى، وتدريبهم على عمليات التفكير السليم المنطقى والناقد، ويمكن إجمال أهمية الدراسة الحالية فى النقاط الآتية :

١ - تقترح الدراسة الحالية برامج إثرائية للمناهج الدراسية بالمرحلة الإبتدائية، كنشاط تعليمى مكمل، والتى تعطى دوراً جديداً لكل من الدراسة والمعلم والتلاميذ.

٢ - تقدم برامج تعليمية تساعد على اكتشاف وزراعة الموهبة لدى تلاميذ المرحلة الابتدائية بما تقدمه من أنشطة ذاتية، تتحدى قدراتهم، وتزيد من مستوى الدافعية لديهم للتعلم واستكشاف الجديد.

٣ ـ تصلح البرامج المقترحة كنشاط تعليمى إضافى علاجى للتلاميذ الذين يعانون من صعوبات فى التعلم أو تأخر دراسى، تأخذ بأيديهم حتى يصلوا إلى مستوى زملائهم المتقدمين دراسياً.

٤ ـ يحقق استخدام التلاميذ للتعلم الذاتى أثناء دراستهم للبرامج المقترحة كثير من الدوافع والحاجات المهمة، ومنها الدافع المعرفى، وتمكنهم من إعادة بناء وتنظيم بطريقة أكثر تكاملاً وأوضح معنى، ويعمق فهمهم للمناهج الدراسية القائمة.

٥ ـ تقدم البرامج الإثرائية المقترحة لتلاميذنا تعليماً تكميليًّا Complementary learning، يقوم بسد النقص فى أنشطة المدرسة وفعاليتها التعليمية، والتى تعجز المدرسة فى تقديمه بسبب كثافة الفصول الدراسية، ونقص الموارد، كما تحقق الأهداف الوجدانية التى لا تستطيع المناهج القائمة تحقيقها مثل تحقيق الذات، وتكامل الشخصية وتنمية الولاء للثقافة وتحقيق مستويات الطموح لدى التلاميذ وتحسين فكرتهم عن أنفسهم وعن الآخرين وتنمية إتجاهاتهم نحو العلم والتعلم.

٦ ـ تقديم البرامج المقترحة لتلاميذنا تعليم إضافى Supplementary Learning، والذى يضيف إلى ما تعلم التلميذ فى المناهج القائمة من معلومات ومهارات.

تحديد مصطلحات الدراسة

يرى لاندال Landal (١٩٩٦) أن الأنشطة المصاحبة للمنهج عبارة عن برنامج أكاديمى مساعد، وأنه قوة مكملة للبرنامج التربوى.

وهو شأنه شأن المواد الدراسية المقررة عبارة عن خبرات يمر بها الفرد، وهى خبرات يؤدى المرور بها إلى تحقيق أهداف التربية وتدعيم أثر التعليم فى حجرة الدراسة، وللنشاط خصائص تميزه عن المواد الدراسية لأن المتعلم عنصر فعال فى اختيار نوع النشاط ووقت وكيفية ممارسته؛ مما يجعل إقباله عليه يتميز بحماس أشد.

ويمكن تصنيف النشاط الإثرائى للمنهج الدراسى إلى:

١ ـ الأنشطة الإضافية للمنهج الدراسى إلى:

Extra Curricular Activities

٢ ـ الأنشطة المنهجية المصاحبة :

Co-Curriclar Activities

٣ ـ البرامج الإثرائية :

يعرف تايور Tayoor (دائرة المعارف الأمريكية، ١٩٧٠) برامج النشاط الإثرائى بأنه :

«البرامج التى تنفذ بإشراف وتوجيه المدرسة، والتى تتناول كل ما يتصل بالحياة المدرسية وأنشطتها المتنوعة ذات الارتباط بالمواد الدراسية».

كما يعرفه جود (Dictionary of Education, 1970) بأنه :

«وسيلة وحافز لإثراء المنهج وإضفاء الحيوية عليه، عن طريق اكتساب المتعلمين لخبرات أساسية تؤدى إلى تنمية معارفهم وقيمهم وإتجاهاتهم بطريقة مباشرة».

وهذا ما تأخذ به الدراسة الحالية.

٤ ـ الوسائط المتعددة :

يطلق عليها الإعلاميات المتعددة أو برمجة الصورة والصوت، وتتضمن الوسائط المتعددة المستخدمة بالكمبيوتر الرسوم المتحركة، التسجيلات، الأصوات، الموسيقى، الألوان، الصور الفوتوغرافية، الصور التخيلية، الرسوم ثنائية وثلاثية الأبعاد، مقاطع من صور الفيديو.

وتتكامل الوسائط السابقة معا باستخدام الكمبيوتر بدرجة، تمكن المتعلم من تناول المعلومات والتفاعل معها من خلال التحكم فى زمن عرضها وخطوات وتتابع العرض من قبل المتعلم، وتعتبر تلك الوسائط منهجاً دراسيًّا مداراً بالكمبيوتر، وليس مجرد معينات للتعلم.

حدود الدراسة

- تشكل العوامل والمتغيرات الآتية حدوداً للدراسة الحالية :

١ ـ تقتصر الدراسة الحالية على اقتراح وصياغة برامج إثرائية للمناهج الدراسية للمرحلة الابتدائية فى المرحلة الحالية، تتبعها باقى مراحل التعليم الجامعى فى أطروحات تالية.

٢ ـ تشتمل البرامج الإثرائية المقترحة على جميع المواد الدراسية بالمرحلة الابتدائية، والتى

يمكن تناولها مثل: اللغة العربية، اللغة الإنجليزية، الرياضيات، العلوم، الدراسات الاجتماعية، التربية الدينية، الأنشطة والمجلات.

٣ ـ تعد البرامج الإثرائية باللغة العربية للمدارس الابتدائية العادية، واللغة الإنجليزية لتلاميذ المدارس الأجنبية.

٤ ـ تستخدم أساليب تنفيذ البرامج الوسائط المتعددة من كمبيوتر، وفيديو منفصلين أو متكاملين، بالإضافة لأفلام الرسوم المتحركة الكرتون (الجرافيك).

٥ ـ تصميم وتبنى الأنشطة الإثرائية المقترحة من قبل باحثى شعبة بحوث تطوير المناهج بالمركز القومى للبحوث التربوية والتنمية بالإضافة لعدد من معلمى المرحلة الابتدائية كخبراء من الميدان، وتنفذ تكنولوجياً على شكل برامج من قبل الفنيين والمتخصصين بالمركز التكنولوجى بوزارة التربية والتعليم.

منهج وإجراءات الدراسة

المنهج:

تتصف البرامج الإثرائية المقترحة فى الدراسة الحالية بأنها ذات طابع توليدى ابتكارى، ولذلك اتبع الباحثون فى مجموعات العمل المختلفة أساليب التفكير الابتكارى فى اقتراح وتحديد معالم تلم البرامج وصياغتها، مثل:

- طريقة عصف الذهن — Brain Storming
- طريقة المشابهات — Symmetrical
- تطوير شجرة الفكرة — Developing an Idea Tree

هذا بالإضافة لاتباعهم المنهج التحليلى الوصفى فى تحليل محتوى المناهج الدراسية المختلفة بالمرحلة الابتدائية من الصف الأول وحتى الصف الخامس.

الإجراءات:

سارت الدراسة الحالية وفقاً للمراحل الآتية:

أولا : تكوين مجموعات العمل وتوزيع المهام :

١ ـ تم تشكيل مجموعات العمل من باحثى شعبة بحوث تطوير المناهج ومعلمى مرحلة ابتدائية بمدرستى بورسعيد بالزمالك ويوسف السباعى بمصر الجديدة.

٢ ـ عقد جلسات تعارف ومناقشة المشروع من حيث أغراضه العامة وفلسفته واقتراح سبل وخطوات التنفيذ فى مقر شعبة بحوث تطوير المناهج ومبنى مدرسة بورسعيد بالزمالك.

٣ ـ أجريت عدة زيارات ميدانية لفريق المشروع للمركز التكنولوجى بالوزارة لتعرف الإمكانات التكنولوجية المتوفرة، وتعرُّف الفريق المنفذ من فنيين ومدخلين للبرامج، وتم تقسيمهم إلى فرق عمل لكل منهج دراسى؛ بحيث يقابلوا نظراءهم من فرق العمل من الباحثين لمناقشة البرامج المقترحة ومتابعة تنفيذها دورياً.

٤ ـ تم تعيين مقررى مجموعات المواد الدراسية المختلفة ومسئولى اتصال، وكذلك رؤساء مجموعات للفنيين بالمركز التكنولوجى.

وبذلك تعرف كل فرد طبيعة المهام المطلوبة وللمجموعة وللمادة الدراسية التى يتبعها وترك حرية عقد اللقاءات الدورية والأسبوعية من حيث الزمان والمكان لأعضاء كل مجموعة ومقررهم.

ثانيا : اقتراح خطط العمل :

١ ـ تم عقد عدة اجتماعات لجميع مجموعات العمل لاقتراح ومناقشة خطط العمل، وضع جدول زمنى بحيث يتم تسليم البرامج الإثرائية للمناهج الدراسية لكل صف دراسى بمعدل شهرى، ابتداءً من الصف الأول الابتدائى، ثم انتهاءً بالصف الخامس الابتدائى.

٢ ـ عقدت كل مجموعة عمل لمنهج دراسى اجتماعات دورية، وتم تحديد التكليف الأول وهو تحليل محتوى المنهج الدراسى المطلوب اقتراح وبناء برامج إثرائية مصاحبة له.

وبعد مناقشة تحليل المحتوى، تم تحديد الأجزاء الصعبة أو الغامضة أو التى تحتاج إلى إلقاء مزيد من الضوء حول مفاهيمها الأساسية لتشكل موضوعات للبرامج الإثرائية.

٣ ـ توالت الاجتماعات الأسبوعية لكل مجموعة لمناقشة التصورات الأولية للبرامج، ومن

خلال جلسات «عصف الذهن»، والمناقشة المفتوحة واستخدام طرق المشابهات وتطوير شجرة الفكرة تم الاستقرار على التصورات النهائية للبرامج الإثرائية للمناهج الدراسية، ثم تجهيزها لتسليمها وفقاً للجدول الزمني للمشروع البحثي.

ثالثا: إنتاج البرامج الإثرائية للمناهج الدراسية باستخدام الوسائط المتعددة:

١ - بعد تسليم البرامج الإثرائية المقترحة للفنيين بالمركز التكنولوجي، تم عقد لقاءات دورية بين مصممي البرامج وفرق تنفيذها لمناقشة تصوراتهم وكيفية التنفيذ ورؤى إخراج العمل، والصعوبات التي قد تظهر أثناء التنفيذ.

٢ - اتضح من واقع المشاهدة والحوار مع المتدربين (معلمين/ موجهين/ مديري مدارس) ظهور بعض السلبيات مما استدعى متابعة الموضوعات للمرة الثانية مع مدخلي البرامج لتعديلها في ضوء المعايير السابق عرضها في الدراسة الحالية؛ بمعنى القيام بالتقويم الذاتي لمصممي البرامج، ومن المقترحات المعروضة تكوين لجنة لتقويم تلك البرامج قبل تعميمها في المدارس.

٣ - وزعت على لجان إنتاج البرامج الإثرائية، وكذلك أعضاء لجان التقويم قائمة بالمعايير التي يجب اتباعها عند اختيار وبناء الأنشطة، وهي:

- تحديد الأجزاء ذات الصعوبة النسبية أو الطبيعة الخاصة، والتي تحتاج إلى عمل أنشطة إثرائية باستخدام الوسائط المتعددة، وذلك في ضوء خبرة الباحثين بشعبة بحوث تطوير المناهج بالمركز القومي للبحوث التربوية والتنمية.
- يركز على عرض المفاهيم العلمية باستخدام المواقف الحياتية، والتأكيد على معالجتها بطريقة تتناسب والمستوى المعرفي للتلاميذ.
- تتضمن الأنشطة المقترحة المفاهيم التي تم اقتراح تضمينها في المناهج المختلفة، والتي تشتمل على:

مفاهيم بيئية - مفاهيم مرورية - مفاهيم سكانية - ترشيد الاستهلاك - الولاء للوطن - احترام الوقت - مفاهيم سياحية - مفاهيم صحية - الإدمان والمخدرات - الاعتماد المتبادل والتعاون ... بحيث يتم تلقائياً، وليس بطريقة قصرية.

- تحديد الأهداف التي ينبغي تحقيقها من خلال الأنشطة المصاحبة.

- تصاغ الأنشطة والمواد التعليمية التي تقترح بصورة متكاملة على شكل وحدات أو محاور.

- تركز الأنشطة على المستقبل وجذب انتباه التلاميذ إلى ممارسة التوقع والتنبؤ العلمي.

- تنمية ولاء الطفل لأسرته ومجتمعه ووطنه.

- تراعي الأنشطة المعدة الفروق الفردية بين التلاميذ وذلك بتنوعها وتعدد مستوياتها.

- تعمل الأنشطة على مساعدة التلاميذ على اكتشاف الجوانب المتفردة والطاقات المختزنة داخلهم.

نتائج الدراسة

تمثل البرامج الإثرائية المقترحة في صورتها النهائية نتائج الدراسة الحالية، والتي غطت موضوعات مختارة من المناهج الدراسية بالمرحلة الابتدائية في جميع تخصصاتها، وذلك للصفوف الدراسية من الصف الأول وحتى الخامس.

وتم تنفيذ البرامج المقترحة في مركز التطوير التكنولوجي تحت إشراف ومتابعة باحثي شعبة بحوث المناهج بالمركز القومي للبحوث التربوية والتنمية، كل فريق في مجال تخصصه، وذلك خلال العام البحثي ٩٦ / ١٩٩٧م. وفقاً للأسس والمعايير والإجراءات السابق عرضها في متن الدراسة الحالية.

كما قادت الباحثة الرئيسة في الدراسة الحالية فريقاً من باحثي شعبة بحوث تطوير المناهج وأساتذة الجامعات، لعقد برامج تدريبية على البرامج الإثرائية باستخدام الكمبيوتر، من خلال نظام التدريب عن بعد، ومؤتمرات الفيديو Video Conference بمركز التطوير التكنولوجي لمعلمي وموجهي ومديري المدارس الابتدائية على تنفيذ البرامج الإثرائية، التي تم إنتاجها خلال الدراسة الحالية. وبداية شمل التدريب البرامج الخاصة بالصفوف الثلاث الأولى من المرحلة الابتدائية وذلك في الفترة من ٦ / ٩ / ١٩٩٧ حتى ١٦ / ١٠ / ١٩٩٧، واشتملت برامج التدريب على المحاور الآتية:

١ ـ فلسفة وأهداف ومعايير إعداد البرامج الإثرائية.

٢ ـ التفاعل بين التلميذ والبرنامج وتحقيق مبدأ التعلم الذاتي.

٣ ـ دور المعلم في البرامج الإثرائية.

٤ ـ تنفيذ البرامج داخل الصفوف.

٥ ـ الوسائط المتعددة لغة القرن الحادى والعشرين.

٦ ـ تهيئة المناخ داخل المدرسة لزيادة فعالية البرامج.

٧ ـ البرامج الإثرائية كوسيلة لتفريد التعلم.

٨ ـ تصميم برامج التعلم لدرجة الإتقان في ضوء فلسفة تكنولوجيا التعليم.

٩ ـ أمثلة تطبيقية من البرامج الإثرائية لكل صف دراسى.

١٠ ـ محتوى البرنامج وعلاقته بالمنهج.

١١ ـ التقويم الذاتي للتلميذ من خلال البرامج الإثرائية.

وبذلك صار الميدان مهيئاً لاستقبال واستخدام البرامج الإثرائية الخاصة بالصفوف الأولى الثلاث من التعليم الابتدائى، وسوف تعقد برامج تدريبية على استخدام البرامج الإثرائية الخاصة بالصفين الرابع والخامس بدءاً من شهر يناير ١٩٩٨ القادم بإذن الله.

وتعرض الدراسة الحالية بعضاً من البرامج الإثرائية المقترحة في جداول مختصرة كنماذج لتعرفها، حيث إنها تأخذ في صورتها التفصيلية حجم مجلدات تزيد عن ألف صفحة، كما أنها تسجل على أقراص ممغنطة لتصبح جاهزة للاستخدام على الكمبيوتر.

نماذج لبعض برامج الأنشطة الإثرائية فى بعض المواد الدراسية

مثال لأنشطة اللغة العربية لمستوى الصف الأول الابتدائى:

الموضوع الأول:

ويشتمل الموضوع الأول على شكل أغنية، يتم تلحينها وغناؤها، وهى تتغنى بمصر وبكونها أحلى البلاد، مع ظهور صور على الشاشة لأهم المعالم الخالدة فى مصر مثل نهر النيل والأهرام وغيرها.

ويهدف الموضوع الأول أن يصبح التلميذ بعد دراسته له قادراً على أن:

١ - يميز بين حرف (ص)، وحرف (س) نطقاً وكتابة.

٢ - ينطق الكلمات نطقاً صحيحاً.

٣ - يتعرف الحروف الآتية: ط، ص، س.

٤ - يكمل الكلمات باستخدام الحرف الناقص.

٥ - يكتب الكلمات التى يطلب منه كتابتها.

٦ - يتعرف استخدام كلمتى: هذا، هذه.

ويمكن توصيف النشاط فى الجدول (١ - أ) التالى:

جدول (١ ـ أ)

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
١	لقاء فى المنزل	ـ حروف : ط، ص، م، س، ف، .. إلخ ـ حوار بين طفل وطفلة يوضح أنهما فى انتظار وصول ضيف. ـ يستقبل الأطفال الأصدقاء ويتعرفوا أهم معالم القاهرة. ـ يتعرفوا أهمية الغذاء وأنواعه أثناء جلوسهم لتناول الطعام.	ـ تظهر الحروف على يمين الشاشات، والحوار بالصورة والحركة على الجانب الأيسر للشاشات. ـ تبرز الكلمات المهمة على الشاشة مع نطقها ثلاث مرات. ـ يظهر فى بداية الطفلين مصطفى وسمر، يعرفان نفسيهما، ويشرحان موضوع النشاط. ـ يذهبان إلى المطار لاستقبال الضيوف. ـ يختار التلميذ الحروف الناقصة لبعض الكلمات من شجرة الحروف ويضع كل حرف فى مكانه من الكلمة.	ـ يستخدم التلميذ المؤشر فى تحريك الحروف التى يختارها يضعها فى مكانها المناسب من الكلمات التى تظهر ناقصة. ـ يعتمد النشاط على التقويم التكوينى بعد كل محور من محاور النشاط الثلاثة. ـ يعتمد التلميذ على بعض شخصيات النشاط، وكذلك بعض الآثار وأنواع من الطعام.

الموضوع الثانى :

يحمل هذا النشاط عنوان «زيارة إلى حديقة الحيوان» ويهدف إلى أن يصبح التلميذ قادراً على أن :

١ ـ يميز بين حرف (ت، ط) نطقاً وكتابة.

٢ ـ يميز بين حرف (د ، ض) نطقاً وكتابة.

٣ ـ يميز بين حرف (ت، ث، ن) نطقاً وكتابة.

٤ ـ يستخدم كلمتى (هذا، هذه) الاستخدام الصحيح.

٥ ـ يكتب الكلمات التى تطلب منه.

٦ ـ يكتب الحرف الناقص من الكلمات التى تعطى له.

٧ ـ يتعرف بعض حيوانات حديقة الحيوان والطعام المناسب لكل منهما.

ويمكن توصيف النشاط فى الجدول (١ ـ ب) التالى :

جدول (١ ـ ب) زيارة إلى حديقة الحيوان

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
١	زيارة إلى حديقة الحيوان : أ ـ بعض الحيوانات تغنى (تحت الحيوانات) . ب ـ الحوار أمام قفص الطاووس. جـ ـ الحوار أمام حظيرة النعامة. د ـ الحوار أمام قفص الثعبان. هـ ـ الحوار أمام بحيرة التمساح وهكذا. السدب ، الزرافة ، القرد ، الأسد	ـ يركز النشاط على الحروف الآتية : (ق، ت، ط، ن، ل، ص، د، و) ـ يركز على كلمتى : هذا، هذه، أثناء عرض كل حيوان. ـ يبدأ بصور لمدخل حديقة الحيوان ويدخل منها الأصدقاء : سمر ومصطفى وحسين وهند. ـ يتوالى ظهور الحيوانات مع حوار حول كل حيوان بين الأصدقاء : الطاووس ـ النعامة ـ ثعبان العرب ـ تمساح ـ ضفدعة ـ الأسد ـ قرد ـ زرافة. ـ يبدأ النشاط بأغنية نحن الحيوانات.	ـ يبرز على الشاشة كل حرف أثناء كل حيوان وكتابة اسمه. ـ تظهر على الشاشة الشخصية المتحدثة، وتشير إلى الحيوان فى كل مرة بكلمة هذا، هذه، مع نطق الجملة مرتين. ـ يكبر كلام ديالوج كل شاشة بالتدرج لتملأ حيز الجزء السفلى من كل شاشة. ـ تظهر كل صورة فى نشاط الكتابة حسب ظهور الصورة، مع ثبات صورة شجرة الحروف فى نشاط الكتابة.	ـ يتخير الحرف الناقص الذى يكون الكلمة المعبرة عن الصورة، ويعيد كتابة الجملة مرة أخرى فى عقب كل صورة لحيوان وديالوج. ـ يصل التلميذ صورة كل حيوان باسمه، مع نطق الاسم. ـ يختار الطعام المناسب لكل حيوان. ـ يكتب التلميذ الحرف الناقص فى كلمات مع نطقها. ـ يختار الإجابة الصحيحة.

مثال للأنشطة الإثرائية فى الرياضيات بالصف الثانى الابتدائى :

تعالج برامج الأنشطة الإثرائية الرياضيات بالصف الثانى الابتدائى موضوعات مهمة، مثل : الطول، الوزن والوقت، وتتم أنشطة البرامج بالاشتراك مع شخصية رئيسية كرتونية

يطلق عليه Mr. Math، وتتم على شكل موضوعات: الضرب، القياس، مسائل كلامية.

ويمكن توصيف تلك الأنشطة فى الجدول (٢) التالى :

جدول (٢) السيد رياضيات Mr. Math

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
١ ٢ ٣ ٤ ٥ ٦ ٧	عمليات الضرب (ثمانى مشاهد). جدول الضرب (ثمانى مشاهد). مسائل كلامية على الضرب. تدريبات عامة على المسائل الكلامية (١١ مشهداً). الوزن (٦مشاهد) الأطوال / قياس الأطوال. الوقت / قياس الزمن (٩مشاهد)	ـ مسائل ضرب من خلال عد أرجل حشرات. ـ جدول الضرب (١) ـ جدول الضرب (٢) ـ جدول الضرب (٣) ـ جدول الضرب (٤) ـ عدد النقاط على الضفادع / ـ مسائل حسابية على الضرب. ـ مسائل كلامية على الضرب. ـ صندوق الكلمات. (مسائل كلامية، جمع، طرح، ضرب). ـ النقل على الأرجوحة. ـ وحدات الوزن ـ أنواع الموازين .. ـ كهف الكنز. ـ قياس الأطوال وأدواته.	ـ تدور المشاهد فى حجرة مكتب مع شخصية عالم أستاذ عبقرينو. ـ حشرات ذات أرجل عديدة، ويحسب التلميذ عددها. ـ بعد انتهاء المشهد والحوار تظهر على الشاشة المسائل حسابياً. ـ أربعة أبواب كل باب يفتح على مستوى من مستويات جدول الضرب. ـ الشخصية الرئيسية عبقرينو يشرح للتلاميذ الصور فى المشاهد ويناقشها معهم. ـ تحل المسائل الكلامية خطوة بخطوة وتضاء كل خطوة تنتهى بإجابة صحيحة. ـ قياس أعمدة وأبعاد الأثاث فى حجرة الأطفال. ـ يظهر على الشاشة أربع بيوت كل بيت يرمز لأسبوع وكل بيت به سبع غرف.	ـ عندما يحصل التلميذ على عشر درجات، يحصل على مكافآت عبارة عن أصوات نارية وأضواء مبهجة. ـ اختيار إجابات صحيحة من بين خاطئة. ـ اختيار مزاوجة بين مجموعتين أ، ب. ـ مع الإجابة الصحيحة يسبح عمر ليصل إلى الكنز. وكل مسألة يجيب عنها تعتبر جزءاً من الكنز.

مثال للأنشطة الإثرائية المقترحة لمادة العلوم للصف الثالث الابتدائى :

يتناول هذا النشاط الأمومة فى عالم الحيوان من خلال عرض بعض الصور التى تتمثل فى رعاية الصغار وحمايته من الأعداد.

ويهدف هذا النشاط أن يصبح التلميذ قادراً على أن :

١ ـ يتعرف صوراً من الأمومة فى بعض الثدييات والطيور.

٢ ـ يميز بين الصور المختلفة للأمومة فى بعض الثدييات والطيور.

٣ ـ يستنتج الصفات العامة للأمومة فى الثدييات والطيور.

(الرعاية ، الغذاء ، والحماية)

٤ ـ يتعرف صور الأمومة فى الأسماك والزواحف.

٥ ـ يستنتج الصفات العامة للأمومة فى الأسماك والزواحف.

٦ ـ تنمية بعض الميول العلمية مثل رعاية بعض الحيوانات.

ويمكن توصيف هذا النشاط فى جدول (٣) التالى :

جدول (٣) : الأمومة فى عالم الحيوان من خلال عرض صور مختلفة (الرعاية، الغذاء، الحماية)

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
١	الأمومة فى عالم الحيوان من خلال عرض صور مختلفة من الأمومة : ـ الرعاية. ـ التغذية. ـ الحماية.	أولا : الأمومة فى الثدييات : أ ـ حماية الصغار فى الثدييات. ـ صغير الفيل بين أرجل أمه. ـ صغير الكنغو فى كيس ببطن أمه. ـ قطعة تحمى صغيرها تهاجم كلب. ب ـ رعاية الصغار فى الثدييات وتغذيتها : ـ بقرة ترضع صغيرها.	يتم من خلال عرض الأمومة فى الثدييات، ثم الطيور والأسماك والزواحف. ويتناول العرض صوراً متحركة عن رعاية الصغار وتغذيتها وحمايتها، ويتمثل ذلك فى طريقة التغذية والطيران، وكيفية	هناك نوعان من التقويم : أ ـ التقويم الذاتى المستمر ويعرض فى نهاية كل من الأمومة فى الثدييات أو الطيور أو الأسماك والزواحف، وهو «أنا اختبر نفسى»، وهو من نوع الاختيار من

تابع جدول (٣) الأمومة فى عالم الحيوان من خلال عرض صور مختلفة
(الرعاية، الغذاء، الحماية)

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
		ـ صغير الفرس يجرى خلف أمه. ـ قطة تنقل صغيرها لمكان آمن. ثانيا: الأمومة فى الطيور: أ ـ رعاية الصغار وتغذيتها. ـ الحمامة تغذى صغارها. ـ الدجاجة تعلم صغارها. ـ طائر اللقلق يطعم صغاره. ـ الإوز تعلم صغارها السباحة فى الماء. ـ الصقر يعلم صغاره الطيران. ـ طائر يهاجم ثعباناً ليبعده عن عشه. ثالثا: الأمومة فى الأسماك تحتفظ بعض الأسماك بالبيض بعد تخصيبه فى مكان آمن إلى أن يفقس مثل سمك القرش والبلطي رابعاً: الأمومة فى الزواحف: تحتفظ بعض أنواع من الأفاعي بالبيض مخصباً فى كيس بفتحة البيض ـ تصنع الترسة بيضها المخصب فى مكان آمن.	الهروب والدفاع عن نفسها وقت الخطر. وبعد كل من الأمومة فى الثدييات أو الطيور أو الأسماك والزواحف يعرض تقويم ذاتي مستمر، أنا اختبر نفسى لمعرفة مدى فهم التلميذ للمعلومات التى درسها، وبعد نهاية النشاط ككل بعض تقويم تجميعى. «اختبر معلوماتك» لمعرفة مدى ما فهمه ومايمكن تطبيقه والاختبارات من نوع الاختبار من متعدد مستخدماً الصور المعبرة عن ذلك.	متعدد مستخدماً الصور المعبرة عن ذلك.

مثال لبرامج الأنشطة الإثرائية المقترحة لمادة الأنشطة والمجالات العملية:

قطار المفاجات:

تشتمل على وحدة إثرائية متكاملة تدور حول قطار المفآجات، الذى يتحرك ويسير فى طريقه فى كل محطة يقف، وتختص كل عربة بمحطة من المحطات، وهى بالترتيب الآتى:

عربة الزهور - عربة الحيوانات - عربة اعرف نفسك - عربة اصنع نفسك بنفسك - عربة اعرف بلدك.. إلخ. كما يتضح فى جدول (٤).

جدول (٤) **قطار المفآجات**

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
١	محطة الزهور	أنواع الزهور (ياسمين - عباد الشمس - عصفور الجنة ..). أنواع أوراق الشجر (الشكل - اللون). رسم وتلوين زهور. خطوات عمل ورد ورق الكوريشة.	تخرج الأزهار من عربة الزهور وتملأ حديقة بالمحطة وتتحرك عدسة الزووم ليتعرف التلميذ الأنواع المختلفة من الزهور وأوراق الشجر. يرسم التلميذ بعض الزهور ويلونها بمساعدة الكمبيوتر. ينفذ التلميذ خطوات صنع ورد من ورق الكوريشة بمساعدة الكمبيوتر. يتدرب على عمل نماذج من تناسق الزهور فى أوضاع مختلفة بمساعدة الكمبيوتر.	يحاكى التلميذ النماذج والخطوات التى يتدرب عليها بمساعدة الكمبيوتر.

تابع جدول (٤) قطار المفاجآت

م	الموضوع	المحتوى	أسلوب تنفيذ النشاط	التقويم
٢	محطة الحيوانات	الأرانب : (يتعرف عليه - يرسمه) - الخرتيت. - الضفدعة. - الفيل. - البطة.	- يرتب التلميذ أجزاء الأرنب ليكون صورة صحيحة، كما تظهر له فى ركن الشاشة. - يرسم الأرنب فى خطوات بسيطة باستخدام الكمبيوتر. - كما فى الأرنب. - كما فى الأرنب. - يلون التلميذ كل رسم	يكون التلميذ صورة حيوانات من مجموعة أشكال مختلفة، وعندما تنجح فى تلوين كل صورة لحيوان بنجاح يبدأ الحيوان فى التحرك محافز للتلميذ.
٣	محطة اعرف نفسك	تعبيرات الوجوه : (غضبان - حزين - فرحان).	- تظهر على الشاشة أشكال تعبيرات الوجوه المختلفة، مع كتابة اسم الشعور باللغة العربية والإنجليزية. - يكون الوجوه الثلاثة ويحدد اسم التعبير المناسب.	- يكون الثلاث وجوه كما تعلمها. - يحدد اسم التعبير مرة بالعربية وبالإنجليزى.
٤	محطة اصنع بنفسك	- طرق صناعة قطار من الورق. - طرق صناعة مركب من الورق. - طرق صناعة ضفدعة متحركة.	- تطبيقات لصناعة نماذج ورقية لقطار، وراكب، ولضفدعة بالاستعانة بالكمبيوتر.	- تطبيقات وتدريبات على المهارات التى تعلمها بمساعدة الكمبيوتر.
٥	معالم سياحية فى مصر	- أبو الهول. - منطقة الأهرام	- يتعرف التلميذ المعالم السياحية فى الصورة.	- يكون التلميذ فى الصورة نفسها للمعالم السياحية وهى مبعثرة الأجزاء.

المراجع

أولا : المراجع العربية :

١ ـ رضا سعد السعيد، «المنهج الإثرائى: رؤية مستقبلية لتطوير مناهج الرياضيات بمراحل التعليم العام»، **المؤتمر العلمى الرابع**، المجلد الأول، ١٩٩١، ص ٣٦٥ ـ ٣٨٧.

٢ ـ عبد الرؤوف محمد بدوى، **دراسة تقويمية لبعض الأنشطة الطلابية بالمرحلة الثانوية العامة على ضوء الأهداف الاجتماعية للتربية**، رسالة ماجستير غير منشورة، كلية التربية، طنطا، ١٩٨٥.

٣ ـ عبد العظيم الفرجانى، **تكنولوجيا المواقف التعليمية**، القاهرة، دار النهضة العربية، ١٩٨٧.

٤ ـ عمر عبد العزيز محمود، سعد عبد الرسول، الأنشطة وأثرها على التحصيل الدراسى لدى طلاب المرحلة الإعدادية، **المؤتمر العلمى الأول لكلية التربية باسوان (التربية والتعليم وتحديات المستقبل)**، ١٩٩٢.

٥ ـ فايز مراد مينا، **«مناهج التعليم العام: دراسة تحليلية»** القاهرة، دار الثقافة ١٩٨٠، ص٢٠٥.

٦ ـ محمد أمين حسن، رؤوف عزمى توفيق، «برنامج مقترح فى التربية الوقائية باستخدام الوسائط المتعددة رؤية مستقبلية لتطوير تدريس العلوم فى المرحلة الابتدائية»، المؤتمر العلمى الأول «التربية العلمية للقرن الحادى والعشرين»، **الجمعية المصرية للتربية العلمية**، المجلد الأول، الإسكندرية، أغسطس ١٩٩٧، ص ١٥٣ ـ ١٧٦.

ثانيا : المراجع الأجنبية :

7- Altken, I.,E., "Computerized communication assessment management: A multi-method approach to skills and field assessment, (by ERIC), 1994.

8- Ansubel, D., P., "Educational psychology: A cognitive View", N.Y.Holt, Rinehart, Winston, 1968.

9- Chagas, I., " Teachers as innovators : A case study of implementing the in-

teractive video disc in a middle school science program, Diss. Abs., Int., A 53 (12), p. 268.

10- Ganguly, I., " Teacher's perspectives on the use and utilization of In structional Television, Dis". Abs, A 56 (2), P. 505.

11- Hergerts,Tom, & Other, "Two applied revisions of a multi-media tool: Assessing attitudes towards computer technology and cultural diversity (by ERIC) 1995.

12- Hovn, H., H.,: "The correspondence between the membership informal working groups and groups formed for spare time.", Zeits Christi for empirische padagoyike, 1978, Vil., 2 (1), PP. 36-61.

13- Marchese, S., R.,: "Allternative and review of the comparison of individual modes of instruction in science," School science and mathematics, December, 1997, pp: 669-703.

14- Miller, L. K.& Others, " A procedure for maiming student progress in oposonalized university course," Journal of Applied Behavior analysis, 1974, pp: 87-91.

15- Newton, C., C., : "Activity participation during high schools: In fluencess of involvement on outcomes", Dissertation Abstracts International A, Vol. 52, No. 7, January, 1992, p. 2493.

16- Papert, S., The children's Machine: Rethinking school in the age of the computer, New York: Basic Books, 1993.

17- Smith, M.,R., "Professional development in elementary science teaching using technology, Dis. Abs, A. 55 (2). p.250.

18- Tayoor, G., " Secondary Education Encyclopedia Americans, New York, American corporation, 1965, p. 68.

19- Walter, S., & Gary, D., "An experimental study of the effect of the presence or Absence of living visual Aids in High school Biology classroom upon attitudes toward science and Biology achievement, Journal of Research in Science Teaching, Vol. 22, No. 7, pp: 619 - 620.

تطوير برمجيات تكنولوجيا الوسائط فائقة التعدد*

د. فرحات فرج فرحات ، د. محمد منير عيسى

المستخلص

ترجع الوسائط فائقة التعدد إلى مجموعة من التكنولوجيات (التقنيات) التى تتعامل معاً (ترتب) بطريقة حديثة لتنظيم وتجهيز المشاركات (الاتحادات) بين العناصر المختلفة لمعلومات الوسائط المتعددة (تجميع الوسائط المتعددة فى إطار واحد).

إن للوسائط فائقة التعدد القدرة على إعطاء (توصيل) حرية جديدة للمستخدمين فى كشف كمية هائلة من معلومات الوسائط المتعددة فى أماكنهم (فى أماكن تواجدهم) وطبقاً لاهتماماتهم.

إن الوسائط المتعددة سوف تؤدى إلى مستويات عالية جداً من تفاعل المستخدم، وبالتالى إلى تأخير قليل جداً (استجابة سريعة) مع نطاق واسع خلال قناة الاتصال.

وهذا البحث يناقش السمات (الميزات البارزة) والتكنولوجيات الخاصة بالوسائط فائقة فى ضوء متطلبات التطبيق.

* النص الأصلى مكتوب باللغة الإنجليزية

نظام للاستنتاج التشريعى مع استرجاع المعلومات *

د. محمد أبوالعلا ، م. تامر محمد ، م. ياسر رمضان

المستخلص

الاستنتاج القانونى هو عملية التفكير التى يقوم بها المحامون عند تطبيق القانون فى قضية جديدة والوصول إلى النتيجة القانونية. إنه من الضرورى أحيانا أن يطبق أو يستنتج من النص التشريعى أو اللائحة التنظيمية. إن مستخدم النظام من الممكن أن يصف حقائق الوقف أو القضية، ويحصل على الإجابة مزودة بالمواد القانونية التى طبقت ولماذا. هذا البحث يقدم نظاماً للاستنتاج التشريعى مع استرجاع لمعلومات لقانون البيئة المصرى، ويأخذ التمثيل الهيكلى كمدخل له، ويقدم النصيحة للمستخدم ليقوم بعمل اللازم تجاه قضيته، أو يقوم النظام بخلق أسئلة لنظام الاسترجاع مستخدماً المعلومات التى حصل عليها من المستخدم، من خلال التعامل المباشر معه مستخدماً الأسئلة المولدة.

* النص الأصلى مكتوب باللغة الإنجليزية

مشروع إنتاج برمجيات محلية للدليل القومى للتنمية التعاونية الزراعية بمصر «دراسة حالة»

د. عطا ابراهيم إمام الألفى *

المستخلص

تقدم هذه الورقة دراسة حالة لمشروع إنتاج برنامج كمبيوتر ذكى لإنشاء دليل قومى للتنمية التعاونية الزراعية بمصر يحقق الاستخدامات التالية:

استخدام تعليمى لتعريف أعضاء وقيادات التعاون بمداخل التنمية التعاونية الزراعية، **واستخدام توجيهى** Advisory Expert System لإيضاح متطلبات وإجراءات التنمية التعاونية الزراعية **واستخدام تقييمى** Scoring System لتعرف المستويات الحالية السائدة فى المجالات المختلفة للتنمية الزراعية وأخيرا **استخدام مستقبلى** للإفادة من نتائج الدليل ومؤشراته.

تركز دراسة الحالة على خطوات تطوير وصيانة وتجميع البرمجيات المنتجة فى الدليل من واقع مفهوم هندسة البرمجيات بمساعدة الكمبيوتر Computer Aided Software Engineering. وتركز أيضا على تحليل النظام بطريقة تحليل النظام الهيكلية Structured System Analysis وصمم النظام بناءً على نموذج التصميم موجه الأهداف Object Oriented Design وإخفاء المعلومات Information Hiding.

لقد تم تصميم برمجيات لميكنة عدد من الأعمال المهمة داخل البرنامج لتسهيل وجودة الأداء ولسرعة الإنجاز خاصة فى مرحلة ما قبل التجميع للبرمجيات.

تم بناء المشروع بالتعاون مع فريق من الخبراء فى المجال تحت إشراف كلية الزراعة بالمنصورة قسم الإرشاد الزراعى، وقيادات التعاون الزراعى، والمعمل المركزى للنظم الخبيرة بوزارة الزراعة، وبتمويل من مؤسسة فريدريش ناومان الألمانية.

---

* المشرف على الفريق المعلوماتى القائم بتنفيذ برنامج الدليل القومى للتنمية التعاونية الزراعية

المقدمة

يتطلب تطوير نظم البرمجيات الضخمة طرقا تمتد حتى مستوى العمليات الصناعية، ولكى تكون تامة الكفاءة يجب أن يكون العمل المطور قائماً على وسائل مبنية على الكمبيوتر، أو ما يعرف ببيئة هندسة النظم القائمة على الكمبيوتر
Computer- Aided Systems Engineering (CASE) [١].

يمكن استخدام تلك الوسائل لميكنة عدد كبير من الأعمال وخاصة تلك الأعمال البسيطة والتى تستهلك وقتا مضنيا عند تنفيذها يدويا. إذا بدت العمليات الضمنية غير مرتبط بعضها البعض فإن الجانب الأعظم من العمل يمكن ميكنته بحيث يصبح مخرج العملية الفرعية مدخلا لعملية أخرى. مما لاشك فيه أن طرق هندسة البرمجيات تدعم المطور بإمكانية ميكنة الأنشطة وتحسن النظرة الهندسية، وتوجه الهدف نحو إنتاج برمجيات آلية على مستوى مواصفات التصميم [٢] إلا أن بعضها قد يشكل عبئا ماديا على مطورى البرامج فى بعض الأوقات أو عدم توافر بعضها فى أوقات أخرى؛ الأمر الذى يجعل تطوير برمجيات محلية تؤدى الغرض نفسه أمراً ضرورياً.

تعد مرحلة التحول الاقتصادى التى تمر بها مصر حاليا حاسمة وفاصلة بين أنماط مختلفة وأيديولوجيات متباينة. ففى عالم متغير ينظر فيه إلى المنظمات بوصفها أجهزة مفتوحة على البيئة، أصبح على المنظمات إن أرادت بقاءً وأداءً وفعاليةً أن تستند إلى عدد من الوظائف المنظمية كالصيانة والدعاية والتدريب والتنسيق والمعلومات والإحصاء والدراسات والبحوث والإرشاد.. إلخ [٣].

انطلاقا من إحدى الندوات المهمة حول الدور الجديد للتعاون الزراعى فى ظل التحرر الاقتصادى، والتى حضرتها قيادات التعاون الزراعى فى مصر ومديرو عموم التعاون الزراعى بمحافظات الجمهورية، تم طرح رؤية للإجراءات التى من شأنها دعم دور التعاون الزراعى فى المراحل الجديدة [٤].

وبناءً على توصيات المؤتمر الثالث للإرشاد الزراعي التي تم مناقشتها وإعلانها خلال الندوة حول الموضوعات التالية [٥] :

- بناء دليل وطني للتنمية التعاونية الزراعية، يؤسس على عديد من المؤشرات الموضوعية، التي تعكس نوع ومدى التنمية الزراعية بكل محافظة في عدة مجالات مختلفة.
- استخدام هذا الدليل في مجال التقييم الذاتي للحركة الزراعية بكل قطاعاتها.
- وأن يعد مرشدا ومرجعا لاحتياجات ومتطلبات تنمية الحركة التعاونية الزراعية.

ونظراً لقيام الفريق المعلوماتي ببناء نظام معلوماتي للجمعيات التعاونية [٦] وبناء نظام خبير لتشخيص آفات وأمراض النبات يخدم التعليم الزراعي بصورة جيدة [٧-٨]، فقد تم وضع خطة عامة لمشروع الدليل من قبل رئيس الفريق البحثي [٩] بجامعة المنصورة، كلية الزراعة، قسم الإرشاد وتكليف الفريق المعلوماتي بالاشتراك مع عديد من الخبراء في المجال التعاوني بوضع خطة عمل خاصة لتطوير مشروع الدليل وتنفيذها، وذلك بتمويل من مؤسسة فريدريش ناومان الألمانية.

سوف نستعرض في هذه الورقة أهم المراحل العلمية التي تم تنفيذها حتى إتمام المرحلة المخططة للمشروع.

مراحل تطوير المشروع

إن إنتاج البرمجيات عبارة عن مجهود فريق متكامل، يختلف حجمه طبقاً لحجم المؤسسة التي تمتلكه، ويكون الفريق مسؤلا عن التطوير الكلي ثم الصيانة في المؤسسة الصغيرة على عكس المؤسسات الكبيرة التي توزع العمل في مختلف المراحل على متخصصين في كل مرحلة. ويعتبر الفريق الذي قام بهذا المشروع من النوع الأول.

وبناء على الخطة المقترحة للمشروع التي أعدها رئيس الفريق البحثي وتمشيا مع الطريقة العلمية لتنفيذه من الناحية المعلوماتية، قام فريق المعلومات برسم هيكل بنائي لخطة العمل كما هي موضحة بالشكل رقم ١ . وتعتمد هذه الخطة على إعداد برامج تنفذ آليا عمليات من شأنها تحقيق الهدف من البرامج الفرعية، وتجرى على بيانات افتراضية لحين التوصل إلى البيانات الحقيقية، التي تتفق على شكلها الهيئة القائمة والمشرفة على المشروع، وبما يناسب رأى الخبراء في المجالات المختلفة. وسوف نستعرض الجوانب المختلفة التي قام بتنفيذها الفريق المعلوماتي.

كروت تحليل النظام والوثائق

برنامج المجالات الرئيسية

برنامج تكويد مستويات التعاون

برنامج تحليل البارامترات

برنامج تحليل المؤشرات

نظام قاعدة بيانات ارتباطية

برنامج حساب قيم المؤشرات

واجهة التعامل مع المستخدم

برنامج تقييم الأداء

قاعدة المعرفة

برنامج إبداء النصح

دمج الوسائط المتعددة

شكل (١) الهيكل البنائى لإنشاء دليل قومى للتنمية التعاونية الزراعية بمصر

٢ـ١ طور المتطلبات Requirements Phase

تختص هذه المرحلة ببلورة احتياجات هيئة المشروع من واقع الأهداف المطروحة، وهى:

١ ـ تحديد مجالات الدليل الرئيسية.

٢ ـ مؤشرات الدليل.

٣ ـ عناصر كل مؤشر، وهى:

اسم المؤشر	ماهية المؤشر	استخدامات المؤشر
معيار المؤشر	بارامترات المؤشر	خطوات حساب المؤشر
إجراءات تحسين المؤشر	العوامل المحددة للمؤشر	

تم بناء النموذج التمهيدى السريع Rapid prototype الذى يعكس العمل والأداء الذى تطلبه هيئة المشروع (مع توضيح الفرق بين هذا النموذج وبين المنتج النهائى) ويحقق المقاييس التالية:

ـ فترة بناء قصيرة.

ـ قابل للتغيير والتعديل بسهولة.

ـ يعكس المفاهيم والأشكال الرئيسية التى يحتاجها مشروع الدليل.

ولضمان جودة البرمجيات التى ستنتج فيما بعد، تم عرض هذا النموذج التمهيدى على الخبراء المختصين لكى تتاح فرصة التعديل والإستفادة من اقتراحاتهم التى قد أخذت فى الاعتبار من قبل، كما لم يغفل مراجعة التقارير المطلوبة مع كل من المستخدم الفعلى للنظام والهيئة العليا للمشروع فى هذه المرحلة.

٢ـ٢ طور المواصفات Specifcations Phase

يتضمن تحليل النظام البنائى (الهيكلى) Structured System Analysis تسعة خطوات رئيسية تعطى فى النهاية مستند المواصفات وتحدد الوسائل التى يجب توافرها لتحقيق ذلك [١٠].

تحتاج هذه الخطوات برمجيات جاهزة أو تصمم خصيصا لتسهيلها وسوف نستعرض بإيجاز هذه الخطوات وما تم إنتاجه محليا لتنفيذ معظمها.

٢ - ٢ - ١ خطوات التحليل البنائى للنظام

تم تنفيذ طريقة التحليل الهيكلى للنظام Structured System Analysis، ويمكن توضيح أهم ما اتخذ فيها بإختصار كالآتى:

١ - شكل تدفق البيانات Data Flow Diagram (DFD) وهو يوضح ماذا يحدث فى المنتج، ويبدأ هذا الشكل موجزاً ويتم التوسع فيه وتدقيقه مرحلياً Step wise refinement .

٢ - تحديد أى الأجزاء يتم برمجته بالكمبيوتر Decide what sections to be computerized?

وخلال هذه الخطوة يتم تحديد الأجزاء التى سيتم برمجتها بحيث تحقق الجدوى من ذلك .

٣ - تخصيص تفاصيل تدفق البيانات Specify the details of the data flow

تخصص التفاصيل من حيث تحديد أى نوع من البيانات يذهب فى مختلف التدفقات .

٤ - تعريف منطق العمليات Define the logic of the processes

يلزم تحديد ما يحدث فى العمليات بدقة بعد تحديد عناصر البيانات . وتعتبر طريقة كود شبه الشفرة Pseudo code أنسب الطرق للتعبير عن ذلك .

٥ - تعريف مستودعات البيانات Define the data stores

يتم فى هذه الخطوة تحديد محتويات مستودع البيانات بالضبط وطريقة تمثيل ذلك المحتوى .

٦ - تعريف المصادر الطبيعية Define the physical resources

عند هذه الخطوة يكون الوصول اللحظى قد عرف بالإضافة إلى شكله وتمثيل سجلات الملفات المزمع إنشاؤها كما يمكن تحديد نوعية الملف (تتابعى أو عشوائى . . . إلخ) .

٧ - تحديد مواصفات المدخل / المخرج Determine the Input/Output Specifications

يمكن عند هذه الخطوة بالتحديد إعداد شاشات الإدخال والإخراج وشكل التقارير ، وذلك لأن الفريق المعلوماتى هو المسئول عن التنفيذ الكامل للمشروع .

٨ - تحجيم النظام Perform the sizing : تعرف عملية التحجيم بأنها عملية تحديد البيانات المطلوبة لتحديد الأجهزة Hardware بالضبط لإجراء البرنامج المنتج للدليل مثل حجم

المدخلات ومعدل تكرار التقارير وطول كل تقرير وحجم وعدد السجلات لكل نوع مما يساعد في حساب كمية التخزين المطلوبة.

٩ - تحديد الأجهزة المطلوبة Determine the hardware requirements

تعد الخطوة الأخيرة في خطوات التحليل البنائي هي تحديد الأجهزة المطلوبة لتنفيذ البرمجيات.

٢ - ٢ - ٢ الوسائل المستخدمة لميكنة التحليل البنائي للنظام

برنامج تحليل النظام آلياً (ASA) Auto System Analysis

- قام الفريق بتصميم هذا البرنامج لتحليل نظام مشروع الدليل القومي بطريقة آلية؛ بحيث يساعد على الاستنباط الآلي للمواصفات التالية:

١ - قاعدة بيانات البارامترات مصنفة تحت مجموعة من المجالات (النماذج) الرئيسية.

٢ - قاعدة بيانات المؤشرات مصنفة تحت مجموعة من المجالات (النماذج) الرئيسية.

٣ - نظام للشرح والنصيحة.

٤ - تحجيم النظام ككل.

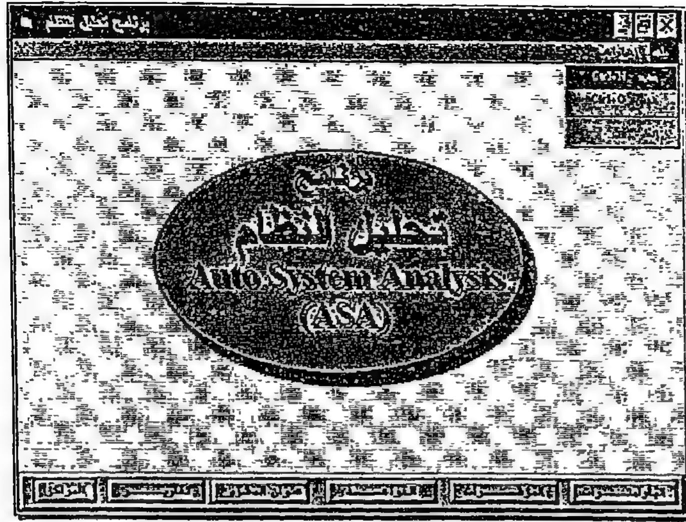
- يسمح هذا البرنامج بالمرونة في التعامل مع مكونات البرنامج وذلك من حيث عمليات الإضافة والحذف والتعديل بما يؤهل إدخالها لمرحلة التصميم والتطبيق للبرنامج الرئيسي.

- يسهل هذا البرنامج عمليات البرمجة، ويوفر الوقت المستخدم فيها، ويقلل من الأخطاء الناتجة أثناءها، كما يسمح بميكنة جزء كبير منها.

- **مكونات البرنامج**

يتكون برنامج (ASA) المصمم لميكنة تحليل نظام التنمية التعاونية الزراعية، كما هو مبين بالشكل (٢) من عدة أجزاء أهمها:

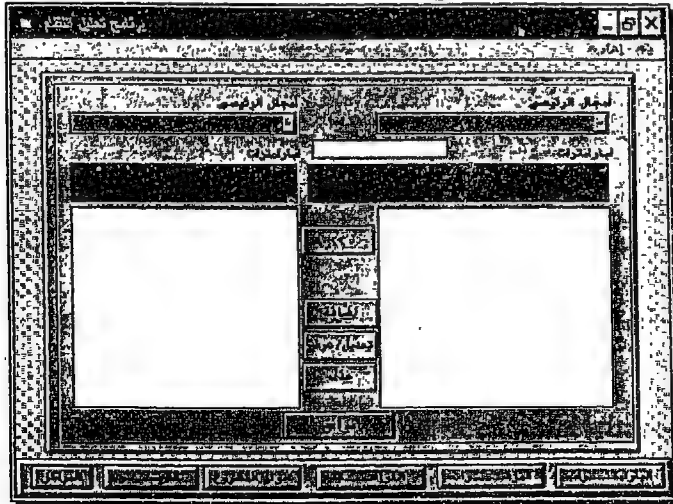
تحليل البارامترات آليا - تحليل المؤشرات آليا - تكوين نظام للشرح آليا



شكل (٢)

أولا ـ برنامج تحليل البارامترات Auto Parameters Analysis

يقصد بالبارامترات حقول البيانات الخاصة بالمجال (نموذج فى دليل التنمية)، والتى تم الاتفاق عليها بواسطة خبراء نظام الدليل القومى، ويتم إعدادها آليا من خلال هذا الجزء من برنامج التحليل الآلى، كما هو موضح بالشكل (٣).



شكل (٣)

يستطيع البرنامج ميكنة المدخلات والمخرجات اللازمة لتكوين قاعدة البيانات، التي تحقق المجالات (النماذج) المتفق عليها كالآتي:

١ - إدخال البارامترات: يتم إدخال البيانات الخاصة بكل بارامتر [المجال - اسم البارامتر - الكود - الوحدة - نوع الحقل - اتساع الحقل - قائمة إختيارات (للاختيار من متعدد) إن وجد] وذلك كما في الأشكال (٤) و(٥) و(٦).

شكل (٤)

شكل (٥)



شكل (٦)

٢ ـ تحديد الملفات وإخراج البيانات اللازمة لتصميم قاعدة البارامترات والتعامل معها وشاشات إدخال البيانات داخل برنامج الدليل القومى.

٣ ـ حساب طول السجل الواحد عن طريق أطوال حقوله.

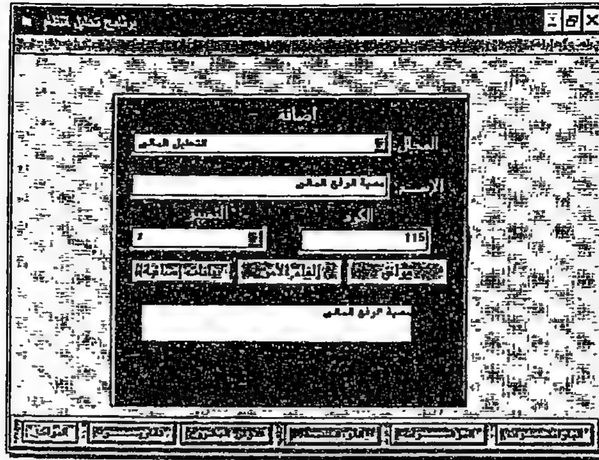
ثانيا ـ برنامج تحليل المؤشرات Auto Indicators Analysis

المؤشر عبارة عن مقياس أداء للجمعية التعاونية فى مجال ما، ويحتوى الدليل على ٢٤٥ مؤشرا. وينفذ البرنامج الوظائف التالية :ـ

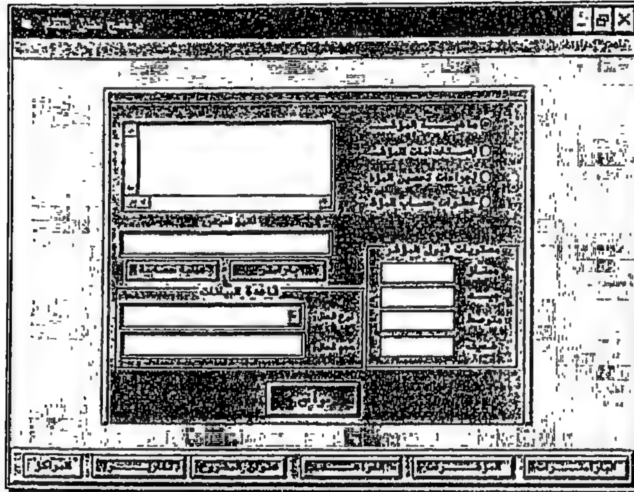
١ ـ إدخال بيانات المؤشرات التالية (المجال ـ إسم المؤشر ـ الكود ـ الوحدة ـ نوع الحقل ـ إتساع الحقل ـ ماهية المؤشر واستخداماته ـ خطوات حساب المؤشر ـ دلالة المؤشر ـ إجراءات تحسينه ـ قانون حساب المؤشر). ويتضح ذلك من الشكلين (٧) و(٨).

٢ ـ تحديد الملفات وإخراج البيانات الخاصة التى تساعد فى تصميم قاعدة بيانات المؤشرات والتعامل معها وحسابها وكذلك شاشات عرضها داخل برنامج الدليل القومى.

٣ ـ حساب طول السجل الواحد عن طريق تجميع أطوال حقوله، وبناء عليه يمكن حساب حجم ملف بيانات المؤشرات التى سيتم التعامل معها وذلك بمعرفة عدد السجلات المدخلة (عدد الجمعيات). ويساعد حساب حجم البارامترات وحجم المؤشرات فى عملية تحجيم النظام.



شكل (٧)



شكل (٨)

ثالثا ـ الشرح Hyper Text (HT)

يصنف هذا البرنامج البيانات التى تم إدخالها فى برنامج تحليل المؤشرات، بحيث يسهل استدعاؤها فى برنامج الشرح HT برنامج النصيحة Advisory ضمن البرنامج الرئيسى (الدليل القومى).

رابعاً ـ تحويل السجلات

عند إجراء التحديث (إضافة / حذف / تعديل) في سجل ما تتغير بالتالى مكونات السجل، ويسبب ذلك مشكلة كبيرة عند إدخال البيانات بكم كبير بواسطة المستخدم. ولمعالجة هذه المشكلة يجب تسكين الحقول القديمة في مواضعها المناظرة (بالحقول الجديدة) بعد التحديث ولهذا السبب تم إنشاء هذا البرنامج.

٣.٢ طور التخطيط Planning Phase

من الأمور الحيوية بالنسبة لمطورى البرمجيات إعداد خطة ذات دقة مناسبة لتكلفة المشروع وكذلك للمدة الزمنية المحتملة لتنفيذه، ويعتبر نموذج Constructive Cost Model (CO-COM) مناسبا [١١] ويعطى نتائج دقيقة، وهو يتلخص في ثلاث خطوات: **الأولى** هي تقدير حجم المنتج (عدد سطور كود المنبع)، **والثانية** تحديد درجة صعوبة أو أسلوب التعامل مع كود البرنامج، **والأخيرة** هي تضمين تأثير بارامترات معينة كدرجة خبرة فريق العمل ومدى إمكانية استخدام وسائل CASE وغيرها. هذا بالإضافة إلى استخدام خطة العمل التى قدمها المعهد الهندسي للكهرباء والإلكترونيات IEEE [١٢] أمكن استخلاص خطة عمل. والجدول التالى يوضح أهم بنود تلك الخطة:

الجدول الزمنى لإعداد برمجيات للدليل القومى للتنمية التعاونية الزراعية (بالأسبوع)

النشاط	مايو				يونيو				يوليو				أغسطس				سبتمبر			
	١	٢	٣	٤	١	٢	٣	٤	١	٢	٣	٤	١	٢	٣	٤	١	٢	٣	٤
١ ـ تحليل المؤشرات	X	X																		
٢ ـ تصميم الشاشات			X	X	X															
٣ ـ تصميم التقارير الخاصة بالمؤشرات				X	X															
٤ ـ تصميم قاعدة البيانات للمؤشرات					X	X	X	X												
٥ ـ تصميم قاعدة النصوص التعليمية				X	X	X	X	X												
٦ ـ إدخال النصوص التعليمية على قاعدة النصوص								X	X	X	X	X								
٧ ـ تصميم قاعدة المعرفة						X	X	X	X	X	X	X	X							
٨ ـ تربيط قاعدة البيانات مع قاعدة النصوص												X	X	X						
٩ ـ تربيط قاعدة النصوص مع قاعدة البيانات وقاعدة المعرفة															X	X	X	X		
١٠ ـ مراجعة النظام مع المختصين																			X	X

٤.٢ طور التصميم Design Phase

نتيجة للتوصل إلى وثيقة المواصفات، يمكن للفريق المعلوماتى معرفة ما يجب عمله بالضبط. وبناء على خطة إدارة المشروع أصبح كل عضو ملما بما سيفعله والمدة المحددة، ولذلك تعرف هذه المرحلة بمرحلة تصميم المشروع. تنال مرحلة التصميم أهمية عالية فى دورة حياة المشروع لما يترتب عليها من صيانة للبرمجيات وهى عملية صعبة، وتستهلك وقتا ومجهودا كبيرين.

وتتكون مرحلة التصميم من خطوتين: **الأولى**، استخراج الأهداف Object وهى الوحدات البنائية الأساسية للمنتج وذلك باستخدام التصميم موجة الأهداف Object-Oriented-Design (OOD) [١٠] كما يمكن تدقيق المنتج إلى مكونات Components.

الثانية، هى التصميم التفصيلى Detailed Design وتتحقق بتدقيق المكونات مرة أخرى وتخصيص الدوال الناتجة إلى وحدات ترجمة Compilation Units ثم العودة إلى تصميم كل دالة مرة أخرى. عند خطوة تصميم كل دالة بالتفصيل، يمكن أن نبدأ طور التطبيق Implementation Phase.

تعطى هذه الخطوات الثقة بأن القطع البرمجية سوف تعمل معا صحيحة كما أنه إذا صمم النظام بطريقة جيدة سيتم الحصول على ميزتين:

١ ـ يمكن بسهولة معرفة ما يجب تغييره عندما يتطلب الأمر التغيير.

٢ ـ أن تؤثر التغييرات على الأجزاء بدرجة كبيرة.

وبالإضافة إلى ما سبق يمكن إستنتاج أن التصميم الجيد يجعل الصيانة أسرع وأكثر أمنا.

٢ ـ ٤ ـ ١ التصميم موجه الأهداف

يتضمن التصميم موجه الأهداف عملية ذات أربع خطوات، هى:

الخطوة الأولى: تعريف المسألة باختصار ما أمكن ويستحسن صياغتها فى جملة واحدة. ويفضل تعريف المشكلة فى جملة واحدة.

«يجب أن يشمل الدليل القومى ثلاث قطاعات رئيسية وهى التصميم والتعليم والنصيحة».

الخطوة الثانية: تطوير استراتيجيات غير رسمية ـ تتابع عام لخطوات تحقق مستند المواصفات تحت القيود المفروضة. للوصول إلي استراتيجية غير رسمية لحل المشكلة يجب التعبير عن المشكلة في فترة واحدة متضمنة القيود المفروضة على النظام.

يشمل الدليل القومي عديداً من المجالات، ويحتوى كل مجال على مجموعة من المؤشرات ولكل مؤشر مجموعة من البارامترات. يتم التعامل مع الجمعيات التعاونية الزراعية بالمحافظات المختلفة ومراكزها وتقيم الجمعية، ثم تصنف على مستوى المركز ومستوى المحافظة ويمكن عرض المجالات، وسبب القصور وطريقة علاج القصور.

الخطوة الثالثة: تعريف الأهداف Objects أو هياكل البيانات والعمليات التي تتم عليها. يعتبر استخلاص الأهداف Objects أو تعريف هياكل البيانات والعمليات التي تنفذ عليها هو قلب التصميم موجه الأهداف. الطريقة المتبعة لذلك هي تعريف الأسماء nouns مع استبعاد ما هو خارج حدود الدليل واستخدام المتبقى كأساس لهيكل البيانات لكل هدف، وتعرف الأفعال verbs وتستخدم كعمليات تتم على الأهداف وذلك كالآتي:

«ينشأ **الدليل القومي** ليطبق على **محافظات الجمهورية** حيث تتكون **المحافظة** من عدة **مراكز**، ويشمل كل مركز مجموعة من **الجمعيات التعاونية الزراعية**. تقوم **الجمعية** بإدخال بياناتها في عدة **مجالات (نماذج)** ويشمل **المجال** عدداً من **المؤشرات**. يقيس **المؤشر مستوى الجمعية** في حدود ماهيته. ويصنف **برنامج الدليل الجمعيات** ويستطيع استخراج **المستوى الهابط** في **المؤشرات** كما يعطى **إمكانية تحسين المستوى** من خلال **نصوص فائقة Hyper Text**. وللدليل **فائدة أخرى** هي **تدريب المختصين** على **المجالات** و**المؤشرات** وطرق **حساب قيمة المؤشر** وبيانات أخرى تخص **المؤشر**. ويستطيع البرنامج **إمداد المستخدم بقاعدة بيانات ارتباطية كمنتج ثانوي** للدليل».

يتم استخلاص الأسماء من الفقرة كما تستخلص الأفعال أيضاً، وتنقح لحذف المتكرر منها، ويتم الحصول على جدول الأهداف والعمليات من واقع الأسماء والأفعال كالتالي:

العمليـــــة	الهـــدف
ـــ	الدليل القومى
سجل بيانات المحافظات	المحافظات
سجل بيانات المراكز	المراكز
سجل بيانات الجمعيات التعاونية	الجمعيات التعاونية
سجل المجالات	المجالات
عملية تقييم	المستوى
سجل المؤشرات	المؤشرات
سجل المتعاملين مع الدليل	المختصين

يمكن زيادة إيضاح العمليات (تدقيقها) وتتم زيادة الدقة حتى نحصل على أكبر مستوى من العمليات لكل هدف.

الخطوة الرابعة: تنفيذ التصميم التفصيلى

يتم فى هذه الخطوة تدقيق مرحلى (على خطوات) لتنفيذ التصميم التفصيلى، ثم البدء فى ذلك التنفيذ.

مفهوم إخفاء البيانات Information Hiding

يعتبر مبدأ إخفاء البيانات من المبادئ المهمة فى عملية التصميم، ويقصد به عزل المفاهيم القابلة للتغيير فى دالة واحدة أو وحدة ترجمة Compilation unit كما يجب ألا تظهر تفاصيل تطبيق هيكل البيانات خارج تلك الوحدة.

٢ ـ ٤ ـ ٢ برنامج التصميم الآلى للنظام Auto System Design (ASD)

الهدف من برنامج التصميم الآلى للنظام (ASD) هو استغلال البيانات المدخلة فى مرحلة التحليل الآلية للنظام بإدخالها للتصميم التفصيلى (البرمجة) للبرنامج الرئيسى (برنامج الدليل القومى)، وذلك لتلافى الأخطاء بأقصى ما يمكن، وتوفير وقت ومجهود المبرمج وسهولة التعديل عند الحاجة، وتحقيق أكبر قدر ممكن من الميكنة لعمليات البرمجة، والتى يصعب عملها يدويًا لضخامة عناصر البرنامج (٥٦٩ بارامتر و٢٤٥ مؤشرا)، وكذلك قابلية تلك العناصر للتغير من قبل الخبراء.

ملفات برنامجى ASA وASD

أولا ـ الملفات الأساسية :

يتم إدخال بياناتها بواسطة محلل النظم في برنامج (ASA)، وهى كالتالي:

١ ـ ملف project.mak : وهو الملف الرئيسى الذى يستدعى باقى ملفات المشروع فى برنامج (ASA) لكى يتم التعامل مع البارامترات والمؤشرات.

٢ ـ ملف project.par : وهو الملف الذى يحتوى على جميع بيانات وتوصيفات البارامترات التى يحتاجها برنامج تحليل النظام (ASA).

٣ ـ ملف project.ind : وهو الملف الذى يحتوى على بيانات ومواصفات المؤشرات، والتى يحتاجها برنامج تحليل النظام (ASA).

٤ ـ ملف project.sub : وهو الملف الذى يحتوى على البيانات الإضافية للمؤشرات والتى يحتاجها برنامج تحليل النظام فى تصميم نظام الشرح (HT) والنصيحة (Advisory).

ثانيا ـ الملفات المساعدة للبرنامج :

وهى ملفات يتم إنشاؤها آليا ـ بناءً على بيانات الملفات الأساسية التى تم إدخالها فى برنامج تحليل النظام (ASA) وهى:

١ ـ ملف Guide.par : يحتوى على توصيفات بيانات البارامترات، والتى يتم استدعاؤها بواسطة البرنامج الرئيسى لملء قاعدة بيانات البارامترات.

٢ ـ ملف Par_db.rpt : يقوم هذا الملف بتوصيف حقول سجل قاعدة البيانات للبارامترات.

٣ ـ ملف Put_par.rpt : يعمل هذا الملف على توصيف حقول سجل قاعدة البيانات للبارامترات.

٤ ـ ملف Get_par1.rpt : يستخدم فى تخصيص القيم اللازم إدخالها لصيغ حساب المؤشرات.

٥ ـ ملف Get_par2.rpt : يساعد فى تصميم برنامج قراءة قيم حقول البارامترات ووضعها فى مصفوفة البارامترات.

٦ ـ ملف Guide.ind : يحتوى على بيانات المؤشرات التى يتم استدعاؤها خلال البرنامج الرئيسى لملء قاعدة البيانات الخاصة بها.

٧ ـ ملف Put_ind.rpt : يساعد هذا الملف فى تصميم برنامج وضع قيم الحقول للمؤشرات والتى قد تم حسابها من قبل البرنامج الرئيسى.

٨ - ملف Ind_db.rpt : يساعد فى توصيف حقول سجل قاعدة البيانات المؤشرات.

٩ - ملف Get_ind.rpt : المساعدة فى تصميم برنامج قراءة قيم حقول المؤشرات ووضعها فى مصفوفة المؤشرات.

١٠- ملف Formula.rpt : تصميم صيغ ومعادلات حساب الموشرات والتى تدخل فى تصميم البرنامج المخصص لذلك.

١١- ملف Convert.rpt : المساعدة فى تصميم برنامج تحويل السجلات فى حالة الرغبة فى تغيير توصيف سجلات قاعدة البيانات من حيث (الإضافة أو التعديل أو الحذف) لأى حقل من حقول أى سجل.

٢-٤-٣ نظام النصح Advisory System

تقاس المؤشرات فى الدليل القومى بمقاييس مختلفة، ترتبط مع بعضها البعض بروابط قد تكون شديدة فى بعض الأحيان ومتوسطة فى أحيان أخرى، كما تختلف من ناحية الحساب الكمى أو الوصفى، وتتدخل خبرة الخبراء فى تقييم بعضها بدرجة كبيرة، الأمر الذى يتطلب معالجة هذه المؤشرات بجزء كبير من الخبرة. وقد عولجت هذه المهمة على ثلاث مراحل:

- إنشاء قاعدة معرفة خاصة Domain Knowledge Base يتم تجهيزها.
- قاعدة معرفة عامة مجهزة من قبل General Knowledge Base [١٣].
- الاستفادة من آلة الاستدلال الداخلى Inference Engine للغة CLIPS [١٤].

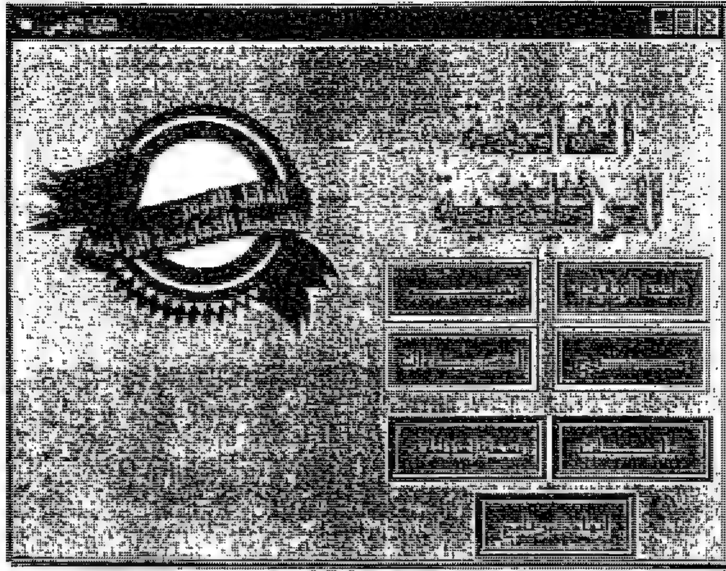
وتعتبر المرحلة الثانية (قاعدة المعرفة العامة) جزءاً مهماً تم بناؤه للمعالجة العامة لقواعد المعرفة الخاصة حيث إنه يحتوى على قواعد ودوال للتعامل مع Classes وما يشتق منها من Instances وما يؤثر فيها من methods [١٣].

٢.٥ طور التجميع Integration Phasa

تعد هذه المرحلة تجميعاً لمنتج برمجيات المشروع فى صورة يمكن بواسطتها تنفيذ الاختبارات الفعلية. ولتوضيح هذه المرحلة سيتم عرض بعض الشاشات الرئيسية مع شرح كيفية التعامل معها بإيجاز.

أ- شاشات البرنامج

يتم من خلالها التفرع إلى جزئيات البرنامج (شكل ٩)، وتتكون من الاختبارات التالية:



شكل (٩)

١ ـ **قاعدة بيانات :** لإدخال بيانات الجمعية .

٢ ـ **تقييم :** لعرض تقييم مؤشرات الجمعية وعرض أسباب القصور والنصيحة .

٣ ـ **شرح :** لشرح الدليل القومى وشرح مؤشراته .

٤ ـ **إحصائيات :** لعمل إحصائيات عن الجمعيات والمراكز والمحافظات المسجل بياناتها فى البرنامج ، وكذلك عمل ترتيب لها حسب تقييمها فى المؤشرات .

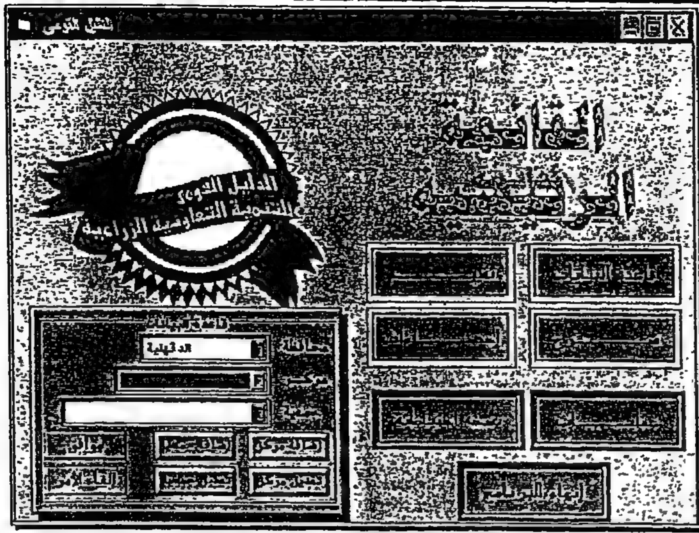
٥ ـ **عمليات :** خاصة بتجميع ملفات إدخال البيانات ـ والتى تم إدخالها فى أكثر من مكان ـ فى ملف واحد .

٦ ـ **نسخة احتياطية :** لعمل نسخة احتياطية لملفات البرنامج .

٧ ـ **إنهاء البرنامج :** للخروج من برنامج الدليل القومى .

ب ـ الدخول فى شاشة قاعدة البيانات

• يتم الضغط على مفتاح قاعدة البيانات فى الشاشة الرئيسية ، فتظهر شاشة فرعية داخل الشاشة الرئيسية ويتم عن طريقها اختيار اسم المحافظة والمركز التابع لها والجمعية التابعة لهذا المركز ، كما فى الشكل (١٠) .



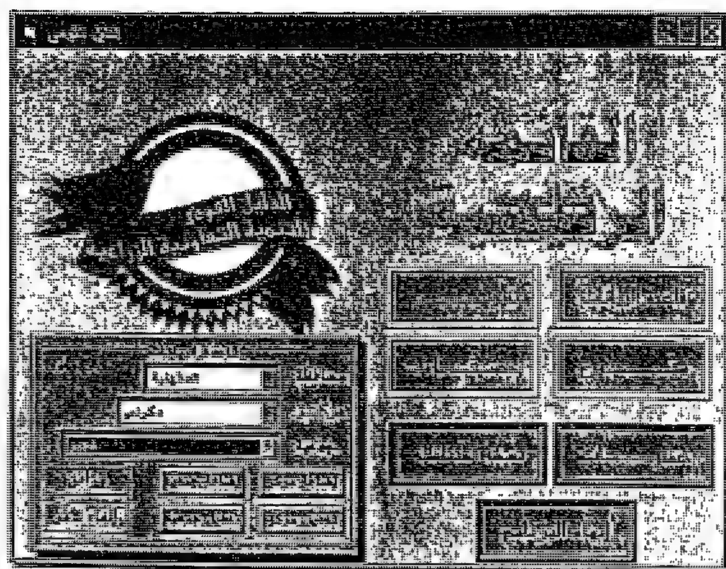
شكل (١٠)

• يمكن إدخال اسم مركز جديد أو جمعية زراعية جديدة، عن طريق الضغط على مفتاح (إضافة مركز - أو إضافة جمعية) كما فى الشكلين (١١) و(١٢).



شكل (١١) شكل (١٢)

• للدخول فى شاشة قاعدة البيانات يتم الضغط على مفتاح (موافق) الموجود فى شاشة إدخال (محافظة / مركز / الجمعية)، كما فى الشكل (١٣).



شكل (١٣)

• فيتم الدخول فى إمكانية إدخال بيانات الجمعية، كما فى الشكل (١٤).

شكل (١٤)

• تحتوى تلك الشاشة على التالى:

١ ـ بيانات الجمعية (المحافظة ـ المركز ـ الجمعية) أعلى الشاشة.

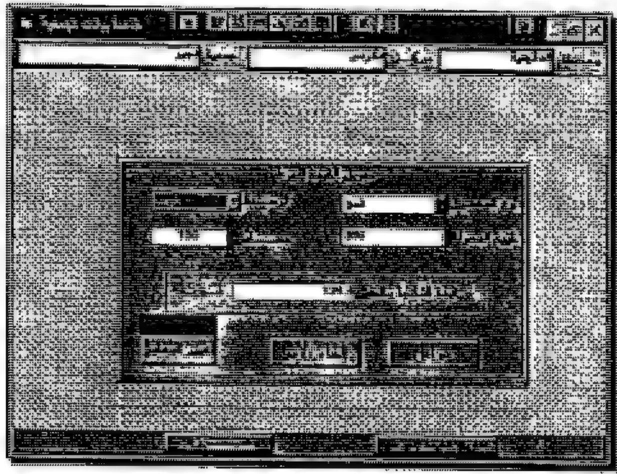
٢ ـ قائمة المجالات: وعن طريقها يتم تحديد المجال المراد إدخال بياناته (نموذج).

٣ ـ البيانات الخاصة بالمجال المحدد، وهى كالتالى:

أ ـ كود البيان ب ـ اسم البيان ج ـ القيمة د ـ الوحدة

٤ ـ مفتاح لتحويل القيمة إلى طن (وذلك لأن بعض الجمعيات تدخل البيانات بوحدات مختلفة مما يتطلب تعديلها حتى يكون أساس التقييم موحدا) كما فى الشكل (١٥).

٥ ـ مفتاح الرجوع للقائمة الرئيسية.



شكل (١٥)

• تكون القيم المدخلة فى شاشة البيانات رقمية أو قيمة اختيارية من اختيارات متعددة كما فى الشكل (١٦).

شكل رقم (١٦)

• تكون القيم المدخلة فى شاشة البيانات رقمية أو قيمة اختيارية من اختيارات متعددة، كما فى الشكل (١٦) :

جـ - شاشة المؤشرات

تحتوى تلك الشاشة على الأجزاء التالية (شكل ١٧).

شكل (١٧)

١ ـ بيانات الجمعية (المحافظة ـ المركز ـ الجمعية) أعلى الشاشة، ويمكن من خلال تلك الشاشة اختيار جمعية أخرى لرؤية تقييم مؤشراتها.

٢ ـ قائمة المجالات: يتم عن طريقها تحديد المجال المراد إدخال بياناته.

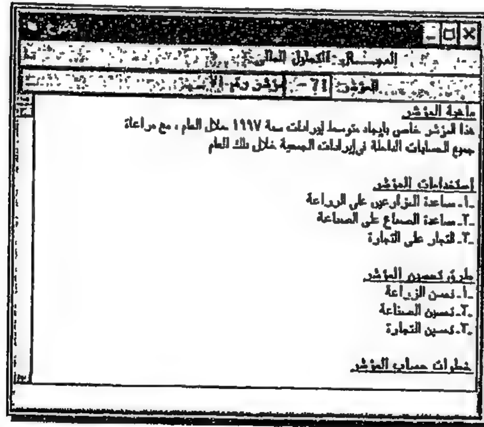
٣ ـ البيانات الخاصة بالمجال المحدد، وهى كالتالى:

أ ـ المؤشر ب ـ درجة المؤشر جـ ـ توضيح د ـ مستوى الترتيب

٤ ـ طرق التحسين (تقييم): لعرض طرق تحسين المؤشر.

٥ ـ الرجوع للقائمة الرئيسية (مفتاح خروج).

• عند احتياج مؤشر ما إلى نصيحة، يتم تحديد ذلك المؤشر من قائمة المؤشرات، ثم يتم الضغط على مفتاح تقييم فتظهر الشاشة شكل (١٨):



شكل (١٨)

المرحلة التجريبية للبرنامج

١ ـ تعتبر الفترة المستغرقة (١٢ أسبوعا) لاستخلاص قائمة المؤشرات وتصميم بطاقاتها وقتا كبيرا مفقودا من فريق العمل المعلوماتى، لولا التفكير فى إنتاج البرمجيات التى تحقق ميكنة جزءاً كبيراً من المراحل المقبلة للمشروع.

٢ ـ يعتبر التنسيق المستمر بين الفريق المعلوماتى (إلحاق مبرمج ومحلل نظم للعمل فى بيئة الدليل القومى) وسيلة مهمة جدا للعرض والمراجعة (ابتداءً من النموذج الأول إلى اختيار إدخال المعلومات) حتى يمكن الوصول بالمنتج إلى درجة تلبية الاحتياج وتحقيق الهدف.

٣ ـ حقق البرنامج التعامل الفعلى مع عدد (٢٣ مجالا) ويندرج تحتها (عدد ٢٠٨ مؤشرات) وبالتالى عدد ٢٧٠ حقلا مختلف القيم (قيمة واحدة ـ إختيار متعدد).

٤ ـ تم بالفعل تعامل عدد ما يقرب من (٣٠ مهندسا زراعيا) من إدارات مختلفة بمحافظة الدقهلية على إدخال البيانات، وظهرت لهم اقتراحات مهمة، تمت إضافتها وصيانة البرمجيات لتحقيقها.

٥ ـ ثبتت إمكانية وفعالية توزيع البرنامج على أجهزة مختلفة للإدخال وتجميع المدخلات فى برنامج واحد (التجربة على مستوى محافظة الدقهلية).

٦ ـ تأخر الفريق المعلوماتى فى إنجاز جزء الشرح فعليًا على بيانات حقيقية، وذلك لتأخر الخبراء فى الصياغة الفعلية لمكونات المؤشرات.

٧ ـ واجه الفريق المعلوماتى صعوبة فى بناء قاعدة المعرفة وخاصة فى بعض المؤشرات؛ وذلك لاختلاف طريقة التقييم للموشر (كمية ـ وصفية).

٨ ـ يفكر فريق العمل فى استخدام الشبكات العصبية فى عملية التقييم لتخفيض جزء كبير من قاعدة المعرفة، ويرجع ذلك لتوفير متجه السمات اللازم لتعليم هذه الشبكات مع توفر برمجياتها [٨].

٩ ـ جمع فريق البحث الرئيسى بيانات من محافظات مختارة، وهى:

أ ـ الإسماعيلية (شرق الدلتا). ب ـ الدقهلية (وسط الدلتا).

جـ ـ البحيرة (غرب الدلتا). د ـ الفيوم (وجه قبلى).

هـ ـ المنيا (وسط وجه قبلى). و ـ قنا (وجه قبلى).

وتم الجمع من عدد (١٦) إدارة تعاونية زراعية بإجمالى حوالى ٦٢٠ جمعية تعاونية زراعية، وذلك لكافة الجمعيات التعاونية متعددة الأغراض والجمعيات المشتركة على مستوى المراكز، وبمعاونة الإدارة المركزية للتعاون الزراعى.

النماذج والمؤشرات

بناء على الأنشطة المطلوبة فى الدليل وما تم فى مراحل التحليل وتحديد المواصفات، تم صياغة نماذج الدليل القومى للتنمية التعاونية الزراعية [١٥].

نماذج الدليل:

أ ـ نموذج (١) دليل **مؤشرات التحليل المالى** (كود ١٠١ إلى كود ١٨٨).

ب ـ نموذج (٢) دليل **مؤشرات التسويق التعاونى** (كود ٢٠١ إلى كود ٢٥١).

جـ ـ نموذج (٣) دليل **خدمات الجمعية التعاونية** (كود ٣٠١ إلى كود ٣٣٢).

د ـ نموذج (٤) دليل **البيانات الإدارية التعاونية** (كود ٤٠١ إلى كود ٤٦٤).

هـ ـ نموذج (٥) دليل **التجهيزات والإمكانات المادية** (كود ٥٠١ إلى كود ٥٦٦).

و ـ نموذج (٦) دليل **الإنتاجية والاحتياجات التعاونية** (كود ٦٠١ إلى كود ٦٤٧).

حـ ـ نموذج (٧) دليل **التنسيق التعاونى** (كود ٧٠١ إلى كود ٧٢٣).

ط ـ نموذج (٨) دليل **الإرشاد والتدريب والخدمات الاجتماعية** (كود ٨٠١ إلى كود ٨٣٢).

ك ـ نموذج (٩) دليل **المشروعات التعاونية** (كود ٩٠١ إلى كود ١٠٤٨).

ل ـ نموذج (١٠) دليل **الدعم المعلوماتى** (كود ٢٠٠١ إلى كود ٢٠٠١٨).

مجالات الدليل ومؤشراته :

يضم الدليل عددا من المؤشرات التى تم حسابها من خلال بنود القياس بعد مناقشتها مع الخبراء، وسوف نقدم هنا فقط (٤) نماذج من مؤشرات كل مجال.

- **مؤشرات التحليل المالى:**

١ - نسبة السيولة. ٢ - نسبة الفائض إلى رأس المال العامل.

٣ - معدل دوران المخزون. ٤ - نسبة الرفع المالى.

- **مؤشرات التمويل والإقراض التعاونى:**

١ - نسبة تحصيل القروض النقدية. ٢ - الملاءمة التمويلية.

٣ - نسبة تحصيل القروض النقدية. ٤ - متوسط تكلفة التمويل المتاح.

- **الخدمات التعاونية الزراعية:**

١ - الخدمات الفنية المقدمة من الجمعية.

٢ - نسبة المساحة المخدومة بالرى من خلال الجمعية (الموسم الشتوى).

٣ - نسبة أطول المراوى والمصارف المطهرة بزمام الجمعية.

٤ - نسبة الإيرادات / التكاليف للخدمات الآلية.

- **القيادات التعاونية:**

١ - مستوى التجديد فى عضوية مجلس الإدارة.

٢ - نسبة إسقاط العضوية عن أعضاء المجلس.

٣ - نسبة الفوز بالتزكية.

٤ - المعامل النسبى لمتوسط حيازة أعضاء مجلس الإدارة.

- **الإمكانيات البشرية والمادية:**

١ - عدد أعضاء الجمعية ذوى الحيازة أكثر من ٥ فدادين.

٢ - نسبة مساحة الفاكهة إلى مساحة الزمام.

٣ ـ عدد سنوات خبرة مدير الجمعية.

٤ ـ عدد الجرارات بالجمعية.

- **التسويق التعاونى الزراعى:**

١ ـ نسبة توفر الخدمات التسويقية.

٢ ـ نسبة الحاصلات المسوقة تعاونيا إلى إجمالى المنتج.

٣ ـ شروط صلاحية المخازن.

٤ ـ نسبة التمويل الذاتى من أموال التسويق.

- **مستلزمات الإنتاج:**

١ ـ مقدرة الجمعية على توفير مستلزمات الإنتاج بسعر أقل من السوق.

٢ ـ نسبة تغطية الجمعية لاحتياجات الأعضاء من التقاوى.

٣ ـ نسبة تغطية الجمعية لاحتياجات الأعضاء من الأسمدة.

٤ ـ الأهمية النسبية لنشاط مستلزمات الإنتاج.

- **ـ الإرشاد الزراعى والتدريب:**

١ ـ التوازن التدريبى للمجموعة الإدارية.

٢ ـ التوازن التدريبى للمجموعات الفنية.

٣ ـ عدد الاجتماعات الإرشادية فى آخر العام.

٤ ـ الاستشارات الفنية الزراعية للإجمالى.

- **الخدمات الإجتماعية التعاونية:**

١ ـ عدد المستفيدين من عمالة التشجير.

٢ ـ عدد المستفيدين من أنشطة عمالة حماية البيئة.

٣ ـ عدد المستفيدين من أنشطة المشاتل.

٤ ـ عدد المستفيدين من أنشطة المشروعات الإنتاجية.

● المشروعات التعاونية:

١ ـ نسبة الاستثمارات فى مجال الأمن الغذائى لإجمالى الاستثمارات.

٢ ـ نسبة الاستثمارات فى مجال الميكنة الزراعية لإجمالى الاستثمارات.

٣ ـ نسبة الاستثمارات فى مستلزمات الإنتاج عن طريق الجمعية المشتركة.

٤ ـ نسبة عائد استثمارات المشروعات.

● ـ التنسيق التعاونى:

١ ـ درجة التنسيق الأفقى الخارجى مع الجمعيات الأخرى.

٢ ـ درجة التنسيق الداخلى بين الجمعية والمنظمات الأخرى.

٣ ـ العلاقة مع المستويات الأعلى (الرؤساء).

٤ ـ مستوى الدور الاجتماعى والتنموى للتعاونيات.

الخلاصة

تقوم دراسة الحالة على تطوير برمجيات محلية فعلية لخدمة مشروع قومى كبير «الدليل القومى للتنمية التعاونية الزراعية بمصر» ؛ مما يعد خدمة للنهوض بكيان كبير فى البيئة المصرية. لقد ساهم فى هذا العمل عدداً كبيراً من خبراء التعاون والإرشاد الزراعى ونظم المعلومات والنظم الخبيرة. وقد تم تحليل النظام بطريقة علمية حديثة، وهى «التحليل البنائى للنظم» كما تم بناء جزءاً كبيراً من البرمجيات لعملية التحليل ذاتها مما يعد انتاجا لـ Shell محلية، تكاد تناهز ما يستخدم فى طرق CASE وذلك بالإمكانات المتاحة.

ومن الدروس المستفادة فى هذا المجال ما يمكن أن نركز عليه فى النقاط المحدودة التالية :

- تحتاج البرمجيات المطورة إلى فريق عمل متكامل، يركز على معالجة الأخطاء أولا بأول لزيادة جودة المنتج.
- يجب التوثيق الجيد حتى يتم التعامل مع المنتج أثناء الصيانة بسهولة ويسر، دون إضاعة للوقت وزيادة التكلفة.
- يجب الأخذ بوسائل هندسة البرمجيات بمساعدة الكمبيوتر وصناعة ما يلزم للبيئة المصرية خاصة أن العقول الواعية المستوعبة للبرمجيات والمطورة لها متوفرة فى مصرنا الحبيبة. كما يجب تشجيع صناعة البرمجيات وعدم الإقلال من قدرها ؛ حتى تصبح لدينا صناعة برمجيات بالفعل.
- تعتبر مشاركة العميل مع المنتج أهم دعائم نجاح البرمجيات، وهذا ما ثبت من خلال هذا المشروع.
- يمكن الاستفادة من الشبكات العصبية فى عمليات التصنيف وخاصة فى السمات الكثيرة الناتجة من قيم المؤشرات الواردة فى الدليل.

المراجع

[1] Ivar Jacobson, Magnus; Christerson Patrik Jonsson and Gunnar Overgaard "Object-Oriented Software Engineering: A Use Case Driven Appoach" ACM Press Addison-Wesley, 1996

[2] Roger S. Pressman "Software Engineering A Practitioner's Apperoach" Mc Graw-Hill, Inc. 1992.

[٣] يحيى على زهران «الحركة التعاونية المصرية بين تنمية المطالب ومطالب التنمية» ندوة المتطلبات المجتمعية للإصلاح الاقتصادى، البعد الغائب فى تنمية الريف المصرى، الجمعية المصرية لعلم الاجتماع الريفى، مؤسسة فريدريش ناومان الألمانية ١٦-١٧ ديسمبر ١٩٩٥م.

[٤] يحيى على زهران « بعض مؤشرات الوضع الراهن للتعاونيات الزراعية فى مصر» ندوة الدور الجديد للتعاون الزراعى فى ظل التحرر الاقتصادى، مؤسسة فريدريش ناومان ١٧ - ١٨ ديسمبر ١٩٩٦م .

[٥] توصيات المؤتمر الثالث للإرشاد الزراعى والتنمية الريفية «مستقبل الخدمات الاستشارية الزراعية فى مصر» مركز الخدمات الإرشادية والاستشارية الزراعية، مؤسسة فريدريش ناومان الألمانية ١٧ - ١٨ ديسمبر ١٩٩٦.

[٦] عطا إبراهيم إمام الألفى «الدعم المعلوماتى للتعاونيات الزراعية بالدقهلية - دراسة جدوى». المؤتمر الثالث للإرشاد الزراعى والتنمية الريفية «مستقبل الخدمات الاستشارية الزراعية فى مصر» مركز الخدمات الإرشادية والاستشارية الزراعية، مؤسسة فريدريش ناومان الألمانية، ١٧ - ١٨ ديسمبر ١٩٩٦م.

[٧] عطا إبراهيم إمام الألفى، محمد كاظم، مها ناصر منصور «نظام خبير لتشخيص وعلاج أمراض وآفات البطاطس» - المؤتمر الثالث للإرشاد الزراعى والتنمية الريفية «مستقبل الخدمات الاستشارية الزراعية فى مصر «مركز الخدمات الإرشادية والإستشارية الزراعية، مؤسسة فريدريش ناومان الألمانية، ١٧ - ١٨ ديسمبر ١٩٩٦م.

[٨] عطا إبراهيم إمام الألفى «استخدام الشبكات العصبية لتطوير نظام خبير لتشخيص وعلاج أمراض وآفات النبات لخدمة التعليم الزراعى «تكنولوجيا التعليم - عدد خاص بالمؤتمر العلمى الخامس للجمعية المصرية لتكنولوجيا التعليم - مستحدثات تكنولوجيا التعليم وتحديات المستقبل.

[٩] يحيى على زهران «موجز تقرير مرحلة إعداد التصور المبدئى ومناقشة الدليل القومى للتنمية الزراعية» جامعة المنصورة - كلية الزراعية - مركز الخدمات الإرشادية والاستشارية - مارس ١٩٩٧م.

[10] Stephen R. Schach "Practical Software Engineering" Boston, IRWIN 1992.

[11] B. W. Boehm, "Software Engineering Economics" Englewood Cliffs, NJ: Prentice Hall, 1981.

[12] IEEE 1987 "A Standard for Software Project Management Plans" IEEE Standard 1058.1-1987.

[13] A.E.E. EL-ALFY and M. KAZEM "Case study: An Expert System Development for Control of Seed Potatoes" Proceedings of the 7th Conference on Computer theory and Application 2-4 Sept. 1997 Alexandria. pp. (1-17 to 1-27).

[14] CLIPS User's Guide, CLIPS Version 6.0 May 28-1993 by Joseph C. Giarratano, Ph. D NASA, Lyndon B. Johnson, Space Center Information Systems Directorate Software Technology Branch.

[١٥] يحيى على زهران «موجز تقرير نتائج المرحلة الأولى من بناء الدليل القومى للتنمية التعاونية الزراعية» - جامعة المنصورة - كلية الزراعة مركز الخدمات الإرشادية والاستشارية الزراعية، أغسطس ١٩٩٧ .

الجزء الخامس

الأوجه القانونية لصناعة البرمجيات في مصر

المسئولية القانونية الناشئة عن إستخدام البرمجيات فى شبكة الإنترنت

د. عزة محمود خليل

المستخلص

تعتبر شبكة الإنترنت من أكبر شبكات المعلومات فى العالم، وقد نشأ عن استخدامها فى مجال البرمجيات مشكلات قانونية، لم تكن تطرأ من قبل على فكر رجال القانون فى العالم. ويتناول البحث هذه المشكلات ومن أهمها حقوق صانعى البرمجيات فى العالم، وهل القانون الحالى يضمن لهم الحماية الكافية وإذا لم يكن كذلك فما مشروع القانون المقترح والذى يمكن أن يحقق التوازن القانونى بين حقوق منتجى البرامج فى العالم ومستخدميها عبر شبكة الإنترنت. كما يتعرض البحث لتجربة الولايات المتحدة الأمريكية فى هذا الشأن، ويعرض بعض مشروعات القوانين التى قدمت للكونجرس لتعديل قانون حق المؤلف فى ضوء تكنولوجيا شبكات المعلومات مثل شبكة الإنترنت.

المقدمة

تعيش البشرية الآن عصر ثورة المعلومات، وغدت الحاسبات وشبكات المعلومات من المفردات المهمة فى حياة الشعوب، وأصبح التحدى المطروح على الساحة العالمية فى الوقت الراهن هو الاشتراك فى أكبر شبكات المعلومات العالمية بقصد الإفادة القصوى مما توفره من إمكانيات هائلة، وباتت وصمة الجهل والتخلف تنتظر من لم يلحق بركب هذا المظهر من مظاهر التطور.

وقد طرح استخدام الحاسبات وشبكات المعلومات مشكلات قانونية باتت تشكل تحدياً لفكر رجال القانون، وكان استخدام البرمجيات على شبكة الإنترنت وكيفية مواجهة المسئولية القانونية التى يستشيرها بالقواعد القانونية التقليدية أخطر هذه المشكلات.

ولما كانت برامج الحاسبات محمية فى معظم بلدان العالم بموجب قانون حق المؤلف، فإن السؤال المطروح للبحث هو: هل هذا القانون كافٍ لحماية حقوق مالكى البرمجيات على شبكة الإنترنت؟ وعلى من يقع عبء تعويض مالكِ البرنامج فى حالة الاعتداء على حقوقه؟

للإجابة عن هذا السؤال، نقسم هذه الورقة البحثية إلى ثلاثة أقسام: نتناول فى القسم الأول المسئولية القانونية الناشئة عن استخدام البرمجيات فى شبكة الإنترنت فى القانون الإنجلو أمريكى، والقسم الثانى نتناول فيه المسئولية القانونية الناشئة عن استخدام البرمجيات فى شبكة الإنترنت فى القانون المصرى، أما القسم الثالث فنخصصه للحلول المقترحة لحل هذه المشكلات.

القسم الأول

المسئولية القانونية الناشئة عن استخدام البرمجيات فى شبكة الإنترنت فى القانون الأنجلو أمريكى

يحمى قانون حماية حق المؤلف فى النظام الأنجلو أمريكى حق المؤلف فى استغلال مؤلفه بالنشر وحق الأداء العلنى وحق إعادة الإنتاج وحق التعديل وحق التحوير، فهل هذه الحقوق تتمتع بالحماية بالنسبة للبرمجيات على شبكة الإنترنت؟

لقد ثار هذا التساؤل فى الولايات المتحدة الأمريكية فى الأوساط التقنية والقانونية، وكانت الإجابة عنه هى: إن حقوق مالكى البرمجيات فى شبكة الإنترنت تنتهك يومياً وباستمرار من قبل مستخدمى الشبكة.

فعلى سبيل المثال إذا أخذنا صورة رقمية موجودة على لوحة إعلانات رقمية موضوعة على شبكة الإنترنت، نجد أن جميع حقوق المؤلف السابق ذكرها يتم الاعتداء عليها، كلما حاول مستخدم مشترك فى الشبكة الإطلاع على هذه الصورة والسبب فى ذلك يرجع إلى أن عملية الإطلاع من جانب المستخدم على هذه الصورة تتطلب قيام المستخدم بنقل هذه الصورة إلى حاسبه الشخصى، وهى العملية التى يطلق عليها **Down loading**.

وإذا أضفنا إلى ذلك أن مستخدم الشبكة ممكن له بعد الإطلاع على هذه الصورة أن يقوم بعملية تعديل لها أو تحوير فيها، فإن هذا بلا شك يعد خرقاً لحقوق مالك هذه الصورة المحمية بموجب قانون حق المؤلف.

بمعنى آخر فإن الطبيعة الرقمية للأعمال الموجودة على شبكة رقمية « Digital » تحتم على المشترك عمل نسخة منها لكى يطلع عليها، وذلك بنقلها إلى حاسبه عن طريق عملية الـ Down loading.

ونظرا لذلك فقد ثارت بهذا الشأن مشكلات قانونية فى الولايات المتحدة الأمريكية، وعرضت عدة قضايا على المحاكم والمعروف أن النظام القضائى الأنجلو أمريكى قائم على نظام السوابق القضائية وبالطبع لم يجد القضاة سوابق يمكن أن يستندوا إليها فى إصدار أحكامهم، فالحالات المعروضة عليهم وليدة التكنولوجيا الحديثة للمعلومات؛ لذلك نجد أن أحكام المحاكم فى هذه القضايا كانت متناقضة رغم تشابه هذه القضايا فعلى سبيل المثال :

أقام الناشرون بمجلة الـ Play Boy الشهيرة دعوى أمام القضاء ضد شبكة Bulletin Board Seystem لأن أحد المستخدمين المشتركين فى الشبكة قام بوضع بعض صور هذه المجلة على شبكة B.B.S، ثم قام مستخدمون آخرون بعمل نقل لهذه الصور لحساباتهم الشخصية Down loading.

وبعد تداول القضية بجلسات المحكمة، صدر الحكم بمسئولية القائمين بالعمل فى شبكة الـ B. B. S، وقالت المحكمة فى حيثيات الحكم :

« إن القائمين بالعمل فى شبكة B. B. S هم المسئولون عن الاعتداء الذى حدث على حقوق الناشرين بهذه المجلة الشهيرة، وليس المستخدم الذى قام بوضع تلك الصور على الشبكة؛ مما يشكل خرقا لحقوق الناشرين فى التوزيع والعرض، ولم يتعرض الحكم لخرق حق الناشرين فى إعادة الإنتاج ولهذا واجه نقداً شديداً من قبل رجال القانون».

قضية أخرى مشابهة قامت شركة سيجا بمقاضاة القائمين بالعمل بشبكة B. B. S لأنهم أتاحوا لمستخدمى الشبكة عدة ألعاب خاصة بشركة سيجا على الشبكة، دون الحصول مسبقاً على موافقة المسئولين بشركة سيجا.

فى هذه القضية وجدت المحكمة أن القائمين بالعمل بشبكة B. B. S اعتدوا على حق شركة سيجا فى إعادة الإنتاج، ولكن المحكمة لم تشر هنا إلى حق سيجا فى التوزيع؛ ولذلك فقد أثار هذا الحكم بدوره نقد رجال القانون.

وأمام التناقض فى الأحكام التى أصدرها القضاء الأنجلو أمريكى فى قضايا مشابهة وأمام صعوبة تعرف الطبيعة المتفردة للاتصالات الرقمية : Digital Communication، وأمام

صعوبة تحديد أى حقوق المؤلف على وجه التحديد تم الاعتداء عليها، وأى خطأ صدر بالتحديد من المعتدى على هذا الحق، اقترح بعض رجال القانون تغيير القانون الحالى لحق المؤلف؛ لأنه لم يعد صالحاً لحماية البرمجيات على شبكات المعلومات، وعلى حد التعبير الشهير لأحد رجال القانون هناك: We need to tailor the copyright law to fit the Internet.

ويرى الرأى الراجح بين رجال القانون فى النظام الأنجلو أمريكى أن حماية مالكى البرامج على شبكة الإنترنت لا تتطلب تغيير القانون الحالى بآخر جديد، وإنما كل ما فى الأمر أن القانون الحالى أصبح يضيق عن استيعاب حماية البرامج على شبكات المعلومات؛ ولذلك فالأمر يتطلب بعض التعديلات التى تتناسب مع هذه التكنولوجيا المتطورة، حتى يمكن توفير الحماية المناسبة لمالكى البرامج على شبكات المعلومات. وعلى حد التعبير لأحد أنصار هذا الرأى: The coat is getting a little tight, there is no need for a new one but the old one needs few alterations to fit the Internet.

ويذهب أنصار الرأى القائل بتعديل القانون إلى أن حق المؤلف الخاص بالتوزيع disttibution يحتاج إلى إضافة ليكون كالآتى:

- إن المؤلف له الحق والسلطة فى توزيع النسخ من مؤلفاته للعامة بالبيع أو أى وسيلة أخرى ناقلة للملكية أو الإيجار أو الترخيص أو الإعارة، واقترح أصحاب هذا الرأى إضافة عنصر الإرسال Transmission.

- يفسر أصحاب هذا الرأى وجهة نظرهم بأن الإرسال عبر الشبكات Network Transmission ليس جزءاً من حق التوزيع طبقاً للقانون الحالى، لأن الإرسال ليس بيعاً ولا نقلاً للملكية ولا ترخيصاً ولا إيجاراً ولا إعارة، وهذه هى الحقوق المحمية التى يتضمنها حق التوزيع طبقاً للقانون الحالى.

كما يرى أصحاب هذا الرأى الراجح أننا إذ نظرنا إلى حق المؤلف فى التوزيع، نجد أنه يحمى الحق فى توزيع النسخ حين تكون النسخة وسيطاً ماديًا أو موضوعيًّا يحمل فكر المؤلف المشمول بالحماية، فهذا هو تعريف النسخة المستقر فى أذهان رجال القانون. أما التوزيع عن طريق الإرسال فهو غير مشمول بالحماية لأن الإرسال Transmission ليس بالوسيط الموضوعى أو المادى الملموس، الذى يحمل فكر المؤلف المشمول بالحماية، ولكنه عبارة عن سيل من الموجات المتلاحقة للواحد والصفر (1&0) والتى تنتقل عبر القنوات اتصال سلكية

أو لاسلكية، فهو بهذا المعنى ليس توزيعاً بالمعنى المعروف الذى يعنى انتقال النسخة مادياً من يد إلى يد أخرى، وإنما المرسل فى الإرسال يحتفظ بالنسخة المرسلة فى الوقت نفسه الذى يرسل فيه نسخة جديدة لمستخدم آخر، الأمر الذى يؤكد معه أصحاب هذا الرأى أن إضافة حق الإرسال إلى الحقوق التى يشملها حق التوزيع يعتبر أمراً حتمياً لتحقيق الحماية الكافية لمالكى البرامج على شبكات المعلومات.

ولقد تم عرض مشروع القانون بإضافة حق الإرسال إلى سائر الحقوق التى يشملها حق التوزيع للدراسة من قبل رجال القانون هناك، والذين أدلوا بآرائهم بشأنه.

ويرى البعض أن إضافة حق الإرسال إلى الحقوق التى يشملها حق التوزيع سيبالغ فى حماية مالكى البرمجيات على حساب منفعة المستخدمين المشتركين فى شبكات المعلومات.

ذلك لأن إضافة هذا العنصر سيجعل مالك البرنامج يتحكم فى حق المستخدم فى الإطلاع أو سماع أى عمل فى شكل رقمى، وهذا يخل بحق المستخدم لأنه إذا كان الإطلاع على كتاب لا يشكل اعتداء على حق المؤلف؛ فكذلك الإطلاع على أى عمل رقمى على شبكة الإنترنت لأنه لا يوجد اختلاف بين الأمرين.

كما يرى البعض أن إضافة حق الإرسال تتناقض مع طبيعة العمل على شبكات المعلومات، والذى يتميز بجعل المعلومات والأعمال الرقمية متاحة على الشبكة لكل من يملك حق الدخول عليها من المستخدمين، وإلا فقدت الشبكة وظيفتها فالنظام المعمول به على شبكة الإنترنت هو أن النسخة الموجودة على الشبكة تكون موضوعة؛ لكى تكون قابلة للنقل لحساب المستخدم من خلال الشبكة العنكبوتية أو الويب، وبناء عليه يرى أصحاب الرأى أن السؤال المطروح يجب ألا يتعلق بإضافة حق الإرسال من عدمه، وإنما يجب أن يكون السؤال المطروح كالآتى:

هل يحتاج المستخدم المشترك فى شبكة المعلومات إلى إذن من مالك البرنامج أو العمل الرقمى digital work؛ لكى يطّلع على منتجاته على الشبكة؟ وكيف له الحصول على هذا الإذن؟ وهل عليه أن يدفع مقابلا لمالك البرنامج للإطلاع على منتجاته؟ وكيف يدفع هذا المقابل خاصة وأن مالك العمل الرقمى قد يكون فى أقصى شمال الكرة الأرضية، بينما يكون المستخدم المشترك فى الشبكة فى أقصى جنوبها؟

كما يرى بعض المعلقين إضافة حق الإرسال ضمن الحقوق التى يشملها حق التوزيع، إلا أن إضافة هذا الحق Transmission تعتبر إضافة نظرية لأن البرامج الموضوعة على الشبكة لم يضعها مالكوها إلا لغرض تمكين المستخدمين من الإطلاع عليها من خلال الشبكة فأين الاعتداء هنا إلى حقوقهم؟

ويرى الرأى الراجح بشأن مشروع هذا القانون أن المسئولية القانونية تنشأ إذا قام مستخدم مشترك في الشبكة بنقل البرامج أو الأعمال الرقمية إلى مستخدم آخر غير مشترك في الشبكة، كما تنشأ بالطبع إذا قام العاملون بالشبكة بوضع برامج عليها، دون الحصول على إذن مسبق من مالكى هذه البرامج.

ومازال مشروع القانون قيد الدراسة والبحث ومازالت الآراء مختلفة بشأنه حتى الآن.

القسم الثانى

المشكلات القانونية الناشئة عن استخدام البرمجيات على شبكة الإنترنت فى القانون المصرى

يقرر القانون الحالى لحق المؤلف فى مصر حقًّا مطلقاً فى استغلال مصنفه بكافة طرق الاستغلال، سواء كان مباشرة بنقله للجمهور أو بصورة غير مباشرة، فتنص المادة السادسة من قانون حماية حق المؤلف على:

«يتضمن حق المؤلف في الاستغلال»:

أولاً: نقل المصنف إلى الجمهور مباشرة بأية صورة، وخاصة بإحدى الصور الآتية:

التلاوة العلنية أو التوقيع الموسيقى أو التمثيل أو العرض العلنى أو الإذاعة اللاسلكية للكلام أو الصوت أو للصور أو العرض بواسطة الفانوس السحرى أو السينما أو نقل الإذاعة اللاسلكية بواسطة مكبر الصوت، أو بواسطة لوحة التليفزيون بعد وضعها فى مكان عام.

ثانياً: نقل المصنف إلى الجمهور بطريقة غير مباشرة بنسخ صور منه، تكون فى متناول الجمهور، سواء يتم ذلك بطريقة الطباعة أو الرسم أو الحفر أو التصوير

الفوتوغرافى أو الصب فى قوالب أو التسجيل أو النسخ أو التثبيت على أسطوانات أشرطة مسموعة أو مرئية، أو بأية طريقة أخرى.

وبدراسة نص المادة السابق، نجد أن المشرع المصرى سبق كثير من مشرعى أكثر الدول تقدماً، فصدر القانون ٣٨ سنة ١٩٩٢ بتعديل قانون حق المؤلف رقم ٣٥٤ سنة ١٩٥٤، فأضاف إلى المادة السادسة ثانياً ضمن صور النقل غير المباشر «أو مرئية أو بأية طريقة أخرى»

وبذلك يكون المشرع قد غطى جميع طرق نقل المصنف إلى الجمهور، فتعبير «أو بأية طريقة أخرى» تعبير واسع مطاط يستوعب كل صور النقل التى أفرزتها التكنولوجيا الحديثة للمعلومات؛ خاصة وأن مشرعنا المصرى قد حمى صراحة مصنفات الحاسب الآلى من برامج وقواعد بيانات وما يماثلها من مصنفات، تحدد بقرار من وزير الثقافة، وذلك بموجب الإضافة التى أدخلها على المادة الأولى من القانون ٣٥٤ سنة ١٩٥٤ بموجب القانون ٣٨ سنة ١٩٩٢ المعدل بالقانون ٢٩ سنة ١٩٩٤.

ولعل مشرعنا المصرى بحمايته لمصنفات الحاسب الآلى صراحة بموجب قانون حماية حق المؤلف، وبإطلاقه صور استغلال المؤلف لمصنفه تعبير «أو بأية طريقة أخرى» يكون قد وفر جهد وفكر رجال القانون فى الجدل القانونى الثائر الآن فى النظام الأنجلو أمريكى، ومدى اعتبار حق الإرسال من الحقوق المحمية ضمن الاستغلال المكفول للمؤلف بموجب قانون حق المؤلف الحالى، والذى يعتبر حق الإرسال بلا شك حقًّا محمياً فى القانون المصرى.

وكما أكد مشرعنا المصرى حق المؤلف فى استغلال مصنفة بأية طريقة من طرق الاستغلال، أكد أيضا بأنه لا يجوز لغيره مباشرة هذا الحق، إلا بعد الحصول على إذن كتابى من صاحب حق الاستغلال المالى للمصنف الأصلى أو خلفائه، على أن يتضمن الإذن طريقة ونوع ومدة الاستغلال «مادة ٥ من القانون ٣٥٤ سنة ١٩٥٤».

وقد أفتى المشرع المصرى مدة حماية حق المؤلف فى استغلال مؤلفه بطول حياته وخمسين سنة لاحقة لوفاته، وقد أخذ المشرع المصرى باستثناء واحد على هذه القاعدة يتعلق بالمصنفات المشتركة، أى المؤلف الذى يشترك فى تأليفه أكثر من شخص، وبموجب هذا الاستثناء تحسب مدة حماية المصنف من تاريخ وفاة آخر من بقى حيا من المشتركين.

وحكمة هذا الاستنثناء واضحة فهو دعوة من المشرع لمزج خبرة الشيوخ بحماس الشباب للاستفادة من مدة حماية أطول، وهى حكمة أملاها المنطق.

وجدير بالذكر أنه بعد انتهاء مدة الحماية، يؤول المصنف إلى الملك العام أو الدومين العام Domain؛ بمعنى أن استغلاله يصبح مشاعاً للجميع، غير حاجة إلى إذن مكتوب من خلف المؤلف، مع ملاحظة أن ما يسقط فى الملك العام أو الدومين العام هو الحقوق المالية دون الحقوق الأدبية، التى لا تنقضى بمرور الزمن مهما طال.

وقد أضاف المشرع المصرى بالقانون رقم ٣٨ سنة ١٩٩٢ مصنفات الحاسب إلى قائمة المصنفات المحمية، وجعل مدة حمايتها عشرين عاماً من تاريخ الإيداع، وهو ما انطوى على مخالفة لاتفاقية برن التى تجعل أقل مدة حماية خمسة وعشرون عاما، وتحظر رهن الحماية القانونية بإجراء شكلى مثل الإيداع، وهو ما تداركه المشرع مؤخراً بالقانون رقم ٢٩ سنة ١٩٩٤ الذى جعل مدة حماية مصنفات الحاسب الآلى كأى مصنف أدبى، تحسب طبقاً للقاعدة العامة فى حساب مدة الحماية.

خلاصة القول فى هذه الجزئية من البحث هى أن مالكى البرامج على شبكة الإنترنت يتمتعون بحماية كاملة طبقاً لإحكام القانون المصرى الخاص بحماية حق المؤلف، والذى نص صراحة على حماية مصنفات الحاسب من برامج وقواعد بيانات، وما يماثلها من مصنفات تحدد بقرار من وزير الثقافة، كما أطلق مشرعنا المصرى حماية استغلال المؤلف لمصنفه بكافة طرق الاستغلال؛ مما يقطع بانبساط مظلة الحماية لتشمل الأعمال الرقمية على شبكة الإنترنت وحق الإرسال الذى أثار جدلاً واسعاً بهذا الخصوص فى النظام الانجلو أمريكى، بل إن إطلاق مشرعنا المصرى حق الاستغلال بقوله أو بأى طريقة أخرى يجعل مظلة الحماية من الاتساع، بحيث تشمل أى طريقة أخرى مستقبلة، يمكن أن يفرزها التطور السريع المذهل لتكنولوجيا شبكات المعلومات، ولعل السؤال الذى ننتظر الإجابة عنه هو:

فى حالة ثبوت اعتداء على حق مالك البرنامج على شبكة الإنترنت، ما الجزاء؟

تقوم حماية المؤلف على طريقين: الطريق الجنائى والطريق المدنى:

أما الطريق الجنائى فالقانون المصرى يعد كل انتهاك لحق المؤلف مكونا لجريمة جنائية، **أما الطريق المدنى** فيأخذ المشرع بمبدأ تعويض المؤلفين المجنى عليهم الذين انتهكت حقوقهم، وهذا ما سنتعرض له بالتفصيل فى القسم الثالث من هذا البحث.

هذا... ويضع المشرع المصرى تحت تصرف مالكى المصنفات المحمية إجراءات تحفظية مقصود بها متابعة قراصنة حق المؤلف؛ بغية كفالة حماية مدنية سريعة فعالة للمؤلفين وهذه

الإجراءات تستهدف مواجهة انتهاكات حق المؤلف، ويمكن التميز طبقاً للمادة ٤٣ بين الإجراءات الآتية:

أ . الوصف التفصيلي للمصنف المحمي على نحو يكشف عن هويته.

ب . الحجز على المصنف الأصلي أو نسخة، كذلك الحال بشأن الأدوات، التي تستعمل نسخة في إعادة نشر المصنف أو عمل نسخ منه.

جـ. حصر الإيراد المحصل من النشر أو العرض بمعرفة خبير يندب لذلك إذا اقتضى الحال وتوقيع الحجز على هذا الإيراد في جميع الأحوال.

كما يعطي المشرع رئيس المحكمة سلطة الأمر بهذه الإجراءات، بناء على عريضة مقدمة من المؤلف أو خلفه في حالة انتهاك الحقوق المالية أو الأدبية، ويكون لرئيس المحكمة الحق في كل الأحوال في تعيين خبير لمعاونة المحضر المكلف بالتنفيذ، كما أن له أن يفرض إيداع كفالة مناسبة.

ويحق للطرف الذى صدر ضده الأمر أن يطعن فيه على الوجه الذى رسمه القانون.

الطريق الجنائى:

يعاقب الاعتداء على المؤلف فى صورته البسيطة بالحبس والغرامة التى لا تقل عن خمسة آلاف جنيه، ولا تزيد عن عشرة آلاف جنيه، أو بإحدى هاتين العقوبتين. وفى حالة العودة تكون العقوبة بالحبس والغرامة التى لا تقل عن خمسة آلاف جنيه، ولا تتجاوز خمسين ألف جنيه.

وفى جميع الأحوال تقضى المحكمة بمصادرة النسخ المقلدة والأدوات المستخدمة للتقليد، وينشر ملخص الحكم الصادر بالإدانة فى جريدة يومية أو أكثر على نفقة المحكوم عليه.

وأجاز المشرع للمحكمة عند الحكم بالإدانة أن تقضى بغلق المنشأة التى استغلها المقلدون أو شركاؤهم فى ارتكاب أحد الأفعال الإجرامية السابقة مدة لا تزيد عن ستة أشهر.

الطريق المدنى:

عندما يتم تكييف الاعتداء على حق المؤلف بأنه خطأ، سواء وقع بحسن نية أو بسوء نية، يكون للمؤلف أو خلفه حق الحصول على تعويضات طبقاً للقواعد العامة فى القانون المدنى

الخاص بالمسئولية التقصيرية، ويثور سؤال هو: على من يقع عبء التعويض؟ وهو ما سنتناوله بالتفصيل فى القسم الثالث من هذا البحث.

القسم الثالث

الحلول المقترحة لحماية مالكى البرمجيات على شبكة الإنترنت

ولعل السؤال الذى يطرح نفسه على بساط البحث هنا هو:

فى حالة ثبوت انتهاك لحقوق البرنامج على شبكة الإنترنت، على من يقع عبء تعويض هذا المؤلف عما أصابه من أضرار؟

هذا السؤال يعد من أكبر الأسئلة التى أثارت جدلاً واسعاً فى الفقه القانونى، وقبل أن نعرض الحلول المقترحة للإجابة عنه، ننوه إلى أنه حين يتم اعتداء على حق مؤلف برنامج موضوع على شبكة المعلومات، نكون أمام أطراف ثلاثة هم:

الطرف الأول: وهو المضرور وهو المؤلف الذى تم الاعتداء على حقه.

الطرف الثانى: هو المستخدم المشترك فى الشبكة والذى اعتدى على حق المؤلف المضرور، وهذا المستخدم المعتدى غالبا ما يكون مجهولا؛ حيث لا توجد الوسائل التقنية الكافية التى تساعد على تحديد شخصية المستخدم المعتدى على سبيل الجزم واليقين.

الطرف الثالث: هو هنا القائمون بالعمل فى شبكة الإنترنت، وهم مقدمو الخدمة للمستخدمين المشتركين فى الشبكة، والتى عن طريقها قام المستخدم بالاعتداء على حق صاحب البرنامج الموضوع عليها.

فمن المسئول عن تعويض المضرور صاحب البرنامج المعتدى على حقوقه؟

هذه المسألة تعتبر من المسائل الخلافية فى الفقه القانونى، ولعل السبب فى هذا الخلاف يرجع إلى صعوبة التوصل للمسئول الحقيقى عن الاعتداء على حق المؤلف، إذا كان هذا المسئول أحد المستخدمين المشتركين فى الشبكة. ولكن بالطبع لا تثور صعوبة ما إذا ثبت أن المسئول الذى اعتدى على حق المؤلف هو أحد القائمين بالعمل فى الشبكة، إذا ما عنَّ له وضع

أى برنامج على الشبكة دون الحصول على موافقة مسبقة من صاحب هذا البرنامج، كذلك إذا أسفر التقدم التكنولوجى عن وسائل يمكن عن طريقها التحقق من تحديد شخص المستخدم المعتدى.

لكن تدق المسألة فى حالة صعوبة الوصول إلى المستخدم المعتدى، ولذلك نجد أن الرأى القانونى هنا قد اختلف، فالبعض يرى أنه إذا كان مالكو البرمجيات على شبكة المعلومات ضحايا أبرياء، فإن مقدمى الخدمة على شبكة الإنترنت يعتبرون مخطئين حسنى النية والمعتدى الأثيم على حق صاحب البرنامج غالباً يعتبر مجهولا، ولا يستطيع التقنيون العاملون بالشبكة الوصول إليه إلا نادراً جداً مع بذل أقصى عناية تقنية ممكنة؛ لذلك فأصحاب هذا الرأى يرون أنه إذا كان على أحد أن يعوض المضرور، فيجب أن يكون هو القائم بالعمل بشبكة المعلومات لأن المستخدم المعتدى هو عميلها، وهو يدفع مقابلا لما تقدمه له من خدمة، ولولا هذه الخدمة ما استطاع أن يعتدى على حق صاحب البرنامج. ومن ثم فعلاقة السببية متوفرة بين الضرر الذى أصاب المؤلف المعتدى على حقه. وبين الخطأ الذى ارتكبته الشبكة بتمكينها عميلها الاعتداء على حق المؤلف، وذلك عن طريق ما توفره له من تسهيلات تقنية تمكنه من هذا الاعتداء على حق مالكى البرامج، ودون أيضا أن تأخذ عليه الضمانات الكافية للالتزام بعدم الاعتداء. ولذلك فإذا كان القائمون بالعمل فى الشبكة يتقاضون أجراً من المستخدمين عن أداء الخدمة التى يقدمونها لهم فإنهم يجب أن يتحملوا المخاطر التى تنشأ عن تقديم هذه الخدمة، وذلك بتعويض صاحب البرنامج المضرور، الذى اعتدى على حقه تطبيقاً للمبدأ القانونى الشهير «الغرم بالغنم».

ويضيف أنصار هذا الرأى أن القول بمسئولية شبكة المعلومات عن تعويض صاحب البرنامج المضرور، سيجعل القائمين عليها يقومون ببذل أقصى عناية ممكنة لتجنب الاعتداء على حقوق مالكى البرامج، وذلك بتوعيتهم وبتحذيرهم، وأخذ الضمانات الكافية عليهم، بالإضافة إلى بذل اقصى عناية تقنية، يمكن أن تمنع الاعتداء على حقوق مالكى البرمجيات.

ويرى البعض الآخر أن القول بمسئولية القائمين بالعمل فى شبكات المعلومات عن الاعتداء الذى يقوم به المستخدمون المشتركون فى الشبكة ينطوى على إخلال بالقاعدة الشهيرة «كل خطأ سبب ضرراً للغير يلزم من ارتكبه بالتعويض»؛ فإذا كان الثابت أن الخطأ الذى اضر بالمؤلف لم يصدر من القائمين بالعمل فى شبكة المعلومات، ولكنه صدر من أحد المستخدمين

أو بعضهم، وكون هؤلاء المستخدمين مجهولين لا يجعل المسئولية تقع على عاتق القائمين بالعمل فى الشبكة، فعلاقة السببية بين الضرر الذى وقع على المؤلف وبين القائمين بالعمل فى الشبكة منتفية، والقول بأن الخدمة التى يقدمها القائمون بالعمل فى الشبكة للمستخدمين هى السبب فى خطأ المستخدم المعتدى، قول ينطوى على قدر كبير من التكلف، كما أنه سيجعل هؤلاء القائمين بالعمل فى الشبكات يعتزلون العمل بشبكات المعلومات، وينتهى عصر الإنترنت وتعود عجلة الزمن للوراء.

ويرى البعض الثالث أن الحل الوحيد لهذه المشكلة يقع على عاتق التقنيين، الذين يجب أن يبحثوا عن الحل من خلال تكنولوجيا أمن المعلومات؛ لأن القانون لم يمنع من قبل المجرمين عن ارتكاب جرائم القتل والسرقة، ولذلك قدمت التكنولوجيا الأقفال الحديثة ونظم الإنذارات المطورة لحماية الأرواح والأموال وهكذا الأمر بالنسبة للبرامج على شبكة الإنترنت؛ ولذلك نجد أن علماء أمن المعلومات قد نشطوا إلى اختراع وسائل تقنية حديثة لحماية أمن المعلومات، مثل: كلمات السر وبصمات الأصابع الرقمية، ويقول أصحاب هذا الرأى إن الحماية الفعالة من خلال تكنولوجيا أمن المعلومات لها ميزات، من أهمها:

١ - إنها ستمنع الاعتداء على حق المؤلف قبل حدوثه، بعكس الاعتماد على القانون الذى لا دور له إلا بعد حدوث هذا الاعتداء بالفعل لإزالة اثاره؛ فتكنولوجيا أمن المعلومات تقوم بالوقاية بينما يقوم القانون بالعلاج وبالطبع الوقاية خير من العلاج.

٢ - إن التشفير ووسائل الحماية التقنية يتيح لأصغر المكتبات والمدارس بالدخول على هذه الشبكات بفاعلية وأسلوب قانونى، كما تتيح السماح بالدخول على المعلومات التى تتطلب دفع مقابل لها، وبذلك توفر حماية كاملة لمنتجى البرامج.

وقد أخذ على هذا الرأى الأخير بأن الحماية التقنية مهددة من قبل قراصنة المعلومات، الذين ينشطون لتكسير الأقفال وفك الشفرات؛ لذلك يرى أنصار هذا الرأى أن اعتمادنا على الحماية التقنية وإهمال الحماية القانونية سيجعلنا ندور مع قراصنة البرامج والمعلومات فى حلقة مفرغة من الفعل ورد الفعل، لأنه بمجرد اختراع وسيلة تكنولوجية لحماية البرامج والمعلومات، ينشط القراصنة ومجرمى المعلومات باختراع أسلوب جديد يمكنهم من اختراق هذه الوسيلة. والغريب فى الأمر أن هؤلاء لا يخجلون مما يفعلون بل على العكس، فهم يتباهون بذكائهم المعلوماتى، الذى مكنهم من اختراق أقوى الحصون التى صممها علماء أمن المعلومات.

لذلك نرى أن الرأى الراجح ينادى بالحماية القانونية الرادعة، ولذلك نجد أن التقنيين أنفسهم أكدوا على ضرورة وجود الحماية القانونية، والدليل على ذلك أن السيد جون بيكيت رئيس جمعية مصنعي الحاسبات بالولايات المتحدة الأمريكية (C. B. E. M. A)، والتى تستخدم مليون ونصف مليون عامل فى مجال الحاسبات ـ ذهب إلى مجلس الشيوخ الأمريكى وطالب بسن القوانين الرادعة لقراصنة المعلومات، وقال قولته الشهيرة «كلما طورنا قفلا ضد خطر قراصنة المعلومات، قام أحدهم بتطوير مفتاح لهذا القفل».

رأينا الخاص :

ونرى أنه إذا كان تحديد شخص المستخدم المشترك فى شبكة الإنترنت الذى اعتدى على حق المؤلف يعتبر أمراً صعباً من الناحية التقنية، إلا أنه ليس من الصعب حصر المكان الذى حدث منه الاعتداء على حق المؤلف، وبالتالى حصر الخطأ فى مجموعة معينة من الأشخاص. هذا المكان قد يكون أبعد بلد من بلدان العالم، وهؤلاء الأشخاص قد ينتمون إلى تشريعات مختلفة ومتباينة ... فالأمر هنا يتطلب انضمام الدول المشتركة فى شبكة الإنترنت؛ حتى لا يضيع حق المضرور من مالكى البرمجيات فى دهاليز اختلاف النظم القانونية.

المراجع

[1] Peter W. Martin, Prospecting the Internet, A.B.A.J., Sept. 1995.

[2] Phillip E. Courtney, Untanglin the Internet, Ent. Sys. J.,March 1995, p. 16.

[3] Brian Livingston, The Mother of All Network; Internet, P.C. COMPUTING, Apr. 1994.

[4] John W. Verity & Robert D. Hof, The Internet: How it Will Change the Way you do Business, BUS. Wk., Nov. 14, 1994.

[5] Alison Schneider, Cruising Cyberspace, WASHINGTONIAN Mag., Dec. 1994.

[6] Stratton Oakmont, Inc. & Prodigy Servs. Co., 23 Media L. Rep. (BNA) 1794, 1798 (N.Y. Sup. Ct. May 25, 1995).

[7] Trotter Hardy, The Proper Legal Regime for "Cyberspace, " 55 U. PITT. L. REV. 993 (1994).

[8] See generally A legislative History Of The Communications Act of 1934, Congressional Hearings and Debates, House and Senate Committee Reports, and articles on the political aspects of the Act).

[9] Rick Ayre & Kevin Reichard, Web Browsers: The Web Untangled, PC MAG., Feb. 7, 1995.

[10] Anne M. Fulton, Cyberspace and the Internet: Who Will Be the Privacy Police?, 3 COMM. LAW CONSPECTUS 63, 64 (1995).

[11] Lan Nguyen. Va. Man's Computer Seized in Internet Lawsuit, WASH. POST, Aug. 13, 1995.

[12] Jerry Berman & Daniel J. Weitzner, Abundance and User Control: Renewing the Democratic Heart of the First Amendment in the Age of Interactive Media, 104 YALEL. J. 1619, 1621 (1995).

[13] Kara Swisher, Getting Rick on the Back of the Internet, Wash. Post, July 24, 1995.

CHAPTER 14

Legal Aspects Originating from Using Software Products Via Internet

Dr. Azza M. Khalil

Abstract

The Internet as one of the biggest information network created many legal problems which haven't been come to legal experts mind before. This paper explains these problems and highlights on one of most important problem among them which is copyright owner all over the world. The paper also answered the question; is the existing copyright law enough? and if not, what is the proposed legislation which could realize the intended palance between copyright owners and users of the internet?. The paper also discusses the U.S.A experience in this matter and the bills which were introduced to the Congress to amend the copyright law in light of world wide network technology such as the internet.

- The original text is written in the Arabic Language.

PART FIVE

LEGAL ASPECTS OF SOFTWARE INDUSTRY IN EGYPT

CHAPTER 13

Production of National Directory For Cooperative Education : Case Study

Dr. Atta E.E. EL-Alfy

ABSTRACT

This technical paper presents a case study to a project of an intelligent computer program production for establishing a national agricultural cooperative development directory which fulfils the following usages: educatianal usage to identify cooperative members and leadeis with agricultural cooperative development entries; advisory expert system usage to claify the requirements and procedures of agricaltural cooperative development; scoring system usage to identify current levels in the different areas of agriculture; and future usage for the benefit from the results and indicators of the directory.

The case study emphasizes the steps of development, maintenance and collecting the produced software in the directory from the perspective of the computer-aided software engineering. Also, the study emphasizes the structured systems analysis, the object-oriented design and information hiding. The software design is geared to computerize a number of important works inside the program to facilitate its performance, quality and speed of the software precollection stage.

This project was conducted in cooperation with a team of experts under the direction of the Faculty of Agriculture, Mansoura University and the Central Laboratory of Expert systems at the Ministry of Agriculture.

- The original text is written in the Arabic Language.

J.C. Smith, "Action Theory And Legal Reasoning" Captus University Pub., 1993.

Hsinchun Chen, et al., "Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval" IEEE Expert V. 8, N. 2, April 1993.

Kagayama Shigeru, "The Fundamentals of Expert Systems on Tort in Japan" http: // www. alw. osaka- u. ac. jp/ ~ kagayama/ Expert/ Expert Tort E.html, September 1995

Katsumi Nitta, et al., "A Legal Reasoning System: New HELIC-II" Proc. of FGGS'94, ICOT, Tokyo, December 1994.

Michael Aikenhad, "Legal Knowledge Based Systems: Some Observations on the Future "Web Journal of Current Legal Issues in association With BlackStone Press Ltd, 1995.

V. Mital and L. Johnson, "Adanced Information Systems for Lawyers, Chapman & Hall, 1992.

Machine: Result:

"يحظر عمل ذلك طبقا للمادة ٥٢ من قانون البيئة

والعقوبة لا تقل عن ١٥٠٠٠٠ جنيه ولا تزيد عن ٥٠٠٠٠٠ جنيه"

The text in the reuslt are generated by the system, and the origin text can be shown upon user request.

CONCLUSION

Our goal is to build knowledge-base system that exploit the knowledge extracted from prescriptive texts, and could use its knowledge base to check situations or cases against provisions of the statutory and regulatory texts. The systems could use attribute of the classes to ask the user about the situation. The questions would be generated by using interrogative expressions with the objects and their relevant attribute names, and applying the syntax rules of the Arabic language for questions and declarative sentences. The system compares the user's answer with the expected response, concludes whether the described situation conforms to or violates the law, and indicate the articles that has been violated.

REFERENCES

Aboul-Ela, et al., "knowledge Representation for Environment Law" 5th International Conference On Artificial Intelligence, ICAIA, Feb. 27-March 3, 1997.

Aboul-Ela, et al., "knowledge Acquisition From Statutary Text" 3rd Conference On Sciences Arabization, Science Arabization Society, Ein-Shams University, March 12-13, 1997.

Anthony G. Francis, et al., "A Comparative Analysis of Case-Based Reasoning and Control-Rule Learning Systems "AAAAi-94 Workshop on Case-Based Reasoning, 1994.

Bernard Moulin and Danial Rousseau, "Automated Knowledge Acquisition From Regulatory Texts" IEEE Expert, V. 7, N. 5, October 1992.

Daniels, J.J. and Rissland, E. L., "A Case Based Approach to Intelligent Information Retrieval "Web Journal of Current Legal Issues, 1995.

David M. Steier, et al., "Combining Multiple Knowledge Sources in an Integrated Intelligent System" IEEE Expert V. 8, N. 3, June 1993.

the semantic and pragmatic knowledge helps increase the search recall, a measure of how well a user retrives all relevant documents [Hsinchun , et al., 1993].

In Information retrieval process, users are interested in text of articles and their provisions; and search terms. The system triggered the topics of subject-area, defined in the concept space, to help the user select the search term and then generates questions based on the attributes that describe the concept in both semantic and pragmatic knowledge. The retrieved information is a summary repesents the meaning of the origin one, which can be viewed upon user request.

RETRIEVAL OF APPLICABLE RULES

The operation of retrieving applicable legal rules differs from simple data base retieval in that it is not simply retrieval by associative mechanisms consisting of the specific legal name of the rule or of letter sequences of key words. This differs from simple data retrieval in that by inputting actual legal facts, a final conclusion can be obtained through automated legal reasoning. Moreover, the pricess by which the conclusion was reached and the material which were used in the production of the conclusion can be accessible for any given stage of the reasoning process [Kagayama, 1995].

When it has been determined whether or not the problematic set of facts fall under the applicable law, and the legal interpretative process is completed, then the legal conclusion is reasoned automatically.

Example:

Automated reasoning for certain situation:

Machine: «ما الموضوع الذى تريد أن تستفسر عنه؟»

User: «السفن»

Machine: «يوجد عدة أنواع من السفن ما النوع المقصود؟»

User: «غير محدد»

Machine: «هل يتم إلقاء الزيت والمزيج الزيتى؟»

User: «لا»

Machine: «هل يتم تصريف الملوثات؟»

User: «نعم»

Machine: «أين يتم تصريف الملوثات؟»

User: «البحر الإقليمى والمنطقة الاقتصادية»

(data type is a pointer to an object of a Case Class type).

Case Class:

Action: e.g. («صرف»),

(«صيد» , «قتل»)

Object: e.g. («المخلفات»),

(«الطيور»)

Object Adjective (multipe values): (e.g. «الصلبة والسائلة والغازية»)
Agent: ,Location: ,Time: Condition: ,Exception: ,Reason: Behaviour:

(All attributes are within the scope of action, and data type is Case Class type except for agent and location it is object class and place class respectively).

- **The Syntactic Knowledge:** It includes the grammar rules of the national language, Arabic. It is organized around syntactic relations between words, such as number agreement. This knowledge is used to generate sentences, either questions or declarative. The system is based on generating multiple relevant questions of certain topics, and the user select from them to navigate to other questions based on the previous one. All these questions are not stored, but the system generate them using semantic, pragmatic and syntactic knowledge. The declarative sentences are generated when the system reason the answer to the user.

Integrating all the knowledge sources is used to comprehend the situation or the case the user wants to describe through an interactive and generic discussion between the system and the user.

CONCEPT-BASED INFORMATION RETRIEVAL

The unstructured nature and volume of textual information make the difficult task of information retrieval even more problematic [Hisinchun, et al. 1993]. Document management systems provide little support for suggesting appropriate search terms or helping users to search terms or helping users to search the document space themselves, using the system's basic pattern-matching capability. Researchers and practitioners have proposed various Artifical Intelligence methods to help users choose search terms and articulate queries. When a document management system incorporate semantic knowledge, users can ask to find related concepts or synonyms. The process of continuously tracing cross-refernecing relationships in

how objects are categorized and how they relate to each other in the world. It represents also the lexical knowledge of the words, and the synonyms or related concepts. This knowledge is used to create semantic constraints to link between words and in inference process. This knowledge can be viewed as a concept space, in which various domain-specific concepts are related.

<u>Example</u>:

(مكبرات الصوت) -- هي -- × (الات وعدد) -- تسبب -- ×(ضوضاء) -- هي -- × (ملوثات بيئة)

(آلات التنبيه) -- هي -- × (آلات وعدد)

(نواتج حرق القمامة) -- موجودة في -- × (مكان حرق القمامة) -- هو -- × (مكان شبه مغلق)

(مكان شبه المغلق) -- هو -- × (مكان عام)

<u>N.B</u>. In the above 'semantic net:: " هي = ISA" which has the property of inheritance.

- **The Pragmatic Knowledge:** It identifies the objects being disussed in the current context, the articles of law and their provisions, and the role of each object within the context. Each article is represented as an instance, object, of a CLASS describes the article of law. This CLASS structure is designed, also, to assist the knowledge engineer through knowledge acquisition which depends on acquisition from text. It, also indentifies cross-referencing ralationships in the articles.

<u>Example</u>:

<u>Article Class:</u>

Article No.: (e.g. A49)

Mode: (obligation or permission (e.g. «حظر»)

Case: (The scope of the Article Mode, its data type is a pointer to an object of a Case Class type.

e.g.(«يحظر صيد أو قتل الطيور»),

(« حظر صرف المخلفات في مجاري المياه»)

Exception: (Exception within the scope of Article Mode,

e.g. «حظر صرف المخلفات في مجاري المياه إلا بعد الحصول على ترخيص»

Time: ,Location: ,Reason:

in the system, by satisfying the clauses of the production rules. The jurisprudentially assumptions inherent in the production rule approach to modeling legal knowledge results in numerous difficulties. How dose one determine whether a situation is within the ambit of a rule? what happens if no clear rule can be found that govern the situation? What if rules conflict? In Common law legal systems these and other problems are, at least partially, resolved by reference to previously decided cases. Attempts have been made to develop production rule systems to reason with cases, through the use of rule induction systems. Case based reasoner can match similar cases with each other. Similarity is only found by testing for the presence or absence of predefined factors. But, The process of analogizing by humans shows that the process involves more than the comparison of presence or absence of predeifined factors. Any similarity perceived between situations depends on the context which the situations are viewed [Michael , 1995].

Systems which reason with statutory (or regulatory) provisions have proved more successful than those involved with case law [V. Mital, 1992]. A statute or requlation may have a complex structure with numerous interwined conditional clauses and cross-references. To apply or reason with a certain statute, it may make sense to represent the statute in a computerized form. Users put to the system a description of the relevant situational facts and obtain from it the answers stating which provisions apply and why.

KNOWLEDGE INTEGRATION

Designing a knowledge-based system with a single knowledge source is certainly easier than integrating multiple knowledge sources. The knowledge sources may use representations that differ from each other and from the representation used by the knowledge-based system itself. Our representation of the knowledge depends on the concept of the knowledge level. The knowledge level, as Allen Newell's terminology [David, et al., 1993], is a way of specifying intelligent systems as agents with goals, knowledge, perceptions, and actions. Agents follow the principle of rationality; that is an ideal knowledge-level agent uses its knowledge to select actions that it believes will result in the attainment of its goal. The search depends on the type of interaction between knowledge sources, which are interdependent and integration requires a great deal of search. One method for integration in this case is generate-and-test: One knowledge source generates candidate solutions, and the other filters the candidates to yield directly usable solutions.

The following list describes how the proposed system repesents and integrates the relevant knowledge sources and how the system uses each type of knowledge:

- **The semantic knowledge:** It provides a context independent semantic net of the words. It includes Semantic features of the object and knowledge about

CHAPTER 12

HYBRID STATUTORY REASONING AND INFORMATION RETRIEVAL SYSTEM

Dr. Magdy Aboul-Ela

&

Tamer Mohamed

&

Yasser Ramadan

ABSTRACT

Legal reasoning is the thinking process adopted by lawyers when they apply legal rules to new case and draw legal conclusion. It is sometimes necessary to apply or reason with a certain statute or body of regulation frequently. A client may put to the system a description of the relevant situation facts and obtain from it the answers stating which provisions apply and why. This paper presents a hybrid statutory reasoning and intelligent retrieval system for environment law in Egypt takes as input either the representative structure of a problem case and output an advise to the user to take a particular course of action regarding his case to achieve some objective or realize some legal rights, or generate a query to the information retrieval system by using information derived from generated and interactive inquiry between the system and the user.

AUTOMATED LEGAL REASONING

Early attempts to automate legal reasoning involved the creation of legal expert systems. Expert systems contain knowledge and reasoning strategies such that the computer can apply the knowledge to problems in order to determine a solution. In an expert system, knowledge is stored in the system in the form of production rules. Reasoning in such production rule systems is a process of applying the rules stored

in the system, by satisfying the clauses of the production rules. The jurisprudentially assumptions inherent in the production rule approach to modeling legal knowledge results in numerous difficulties. How dose one determine whether a situation is within the ambit of a rule? what happens if no clear rule can be found that govern the situation? What if rules conflict? In Common law legal systems these and other problems are, at least partially, resolved by reference to previously decided cases. Attempts have been made to develop production rule systems to reason with cases, through the use of rule induction systems. Case based reasoner can match similar cases with each other. Similarity is only found by testing for the presence or absence of predefined factors. But, The process of analogizing by humans shows that the process involves more than the comparison of presence or absence of predeifined factors. Any similarity perceived between situations depends on the context which the situations are viewed [Michael , 1995].

Systems which reason with statutory (or regulatory) provisions have proved more successful than those involved with case law [V. Mital, 1992]. A statute or requlation may have a complex structure with numerous interwined conditional clauses and cross-references. To apply or reason with a certain statute, it may make sense to represent the statute in a computerized form. Users put to the system a description of the relevant situational facts and obtain from it the answers stating which provisions apply and why.

KNOWLEDGE INTEGRATION

Designing a knowledge-based system with a single knowledge source is certainly easier than integrating multiple knowledge sources. The knowledge sources may use representations that differ from each other and from the representation used by the knowledge-based system itself. Our representation of the knowledge depends on the concept of the knowledge level. The knowledge level, as Allen Newell's terminology [David, et al., 1993], is a way of specifying intelligent systems as agents with goals, knowledge, perceptions, and actions. Agents follow the principle of rationality; that is an ideal knowledge-level agent uses its knowledge to select actions that it believes will result in the attainment of its goal. The search depends on the type of interaction between knowledge sources, which are interdependent and integration requires a great deal of search. One method for integration in this case is generate-and-test: One knowledge source generates candidate solutions, and the other filters the candidates to yield directly usable solutions.

The following list describes how the proposed system repesents and integrates the relevant knowledge sources and how the system uses each type of knowledge:

- **The semantic knowledge:** It provides a context independent semantic net of the words. It includes Semantic features of the object and knowledge about

CHAPTER 12

HYBRID STATUTORY REASONING AND INFORMATION RETRIEVAL SYSTEM

Dr. Magdy Aboul-Ela

&

Tamer Mohamed

&

Yasser Ramadan

ABSTRACT

Legal reasoning is the thinking process adopted by lawyers when they apply legal rules to new case and draw legal conclusion. It is sometimes necessary to apply or reason with a certain statute or body of regulation frequently. A client may put to the system a description of the relevant situation facts and obtain from it the answers stating which provisions apply and why. This paper presents a hybrid statutory reasoning and intelligent retrieval system for environment law in Egypt takes as input either the representative structure of a problem case and output an advise to the user to take a particular course of action regarding his case to achieve some objective or realize some legal rights, or generate a query to the information retrieval system by using information derived from generated and interactive inquiry between the system and the user.

AUTOMATED LEGAL REASONING

Early attempts to automate legal reasoning involved the creation of legal expert systems. Expert systems contain knowledge and reasoning strategies such that the computer can apply the knowledge to problems in order to determine a solution. In an expert system, knowledge is stored in the system in the form of production rules. Reasoning in such production rule systems is a process of applying the rules stored

REFERENCES

1- Michael A. Harrison, "The Essential Elements of Hypermedia", Multimedia Systems and Applications, Acdemic Press Ltd., 1995.

2- Wendy Hall, "Hypermedia Tools for Multimedia Information Management", Multimedia Systems and Applications, Academic Press Ltd., 1995.

3- Rob Walters, "Computer - Mediated. Communications: Multimedia Applications", Boston-London, Artech House Inc., 1995.

4- Kaj Cronbaek and Randall H. Trigg, "Hypermedia System Design Applying the Dexter Modé", Communications of the ACM,(Feb. 1994.)

5- Kaj Gronbaek, Jens A. Hem, Ole L. Madesen and Lennert Sloth, "Systems: A Dexter-Based Architecture", Communications of the ACM (Feb. 1994) pp. 65-74.

6- John J. Leggett and John L. Schnase, "Dexter With Open Eyes", Communications of the ACM, (Feb, 1994), pp 77-86.

7- E. Rivtin, and et als, "Navigating in Hyperspace: Designing a Structure-Based Toolbox", Communications of the ACM, (Feb. 1994,) pp 87-96.

8- Gerard Salton and et als, "Automatic Structuring and Retrieval of Large Text Files", Communications of the ACM, (Feb.1994), pp 97-108.

NO.	Criteria
1	Multimode rule.
2	Object rule.
3	Scripting rule.
4	Multi-user rule.
5	Scalability rule.
6	Interoperability rule.
7	Hyperlink / hyperview rule.
8	Technology independence rule.
9	Extensibility rule.
10	Multilingual rule.
11	Performance support rule.
12	Standards rule.

Table (2). The twelve criteria for hypermedia systems.

Notecards is a system to help people organize their ideas. The users of Note Cards are assumed to be authors, designers, and researchers. On the other hand, hypermedia systems are envisioned as tools for building large industrial - strength performance - enhancing systems. All but one of the Halasz criteria are explicity or implicitly covered in the underlying rules. An apparant exception is his rule on "virtual structures".

Virtual structures are an attempt to overcome the static nature of the basic Note Cards model. A user of Note Cards must map his or her ideas into individual units which are stored one to a card. Each card gets a title and is placed into a "filebox". This means that the user must predefine a static stucture and Halasz calls this the "problem of premature organization".

of being purchased and used in significant quantities. For example, the X window System has become a market standard since all the important hardware manufacturers offer implementations of it and all the major software vendors now write user interface to conform with it. Instead of being voted upon with a show of hands by a standards committee, market standards are determined by the vote of dollars in the market place.

Hypermedia systems must comply with and support relevant formal and market standards.

Unfortunately, multimedia and hypermedia are still such a young set of technologies that the standards process for them has not yet run its course.

However, a few standards such as JPEC, MPEG, unicode, MIDL, Hy Time, and MHEG have emerged. Such standards must be supported in hypermedia systems and not replaced by equivalent proprietary methods.

5. CONCLUSIONS

Industrial strength applications are built today with tools such as 4GLs and forms pakages. It is belived that hypermedia systems as described herein can dramatically alter the process and speed of building such applications as well as the sophistication and usability of those applications.

Twelve criteria have been offered by which to judge putative hypermedia systems, refer to Table (1). Not all of these criteria have equal weight.

Rules 1,3 and 7 are essential and unchallenged. Rule 2 is largely agreed upon but other approaches may yet prove viable.

Comparing these 12 rules with the work of Halasz et al, on Note Cards in which he lists seven fundamental issues for hypermedia. The two approaches are very different due to the underlying assumptions differ radically.

readily one's location among them at any time, and to return easily to any prior location or a navigated path.

4.8. The Technology Independence Rule:

A hypermedia system should provide independence from any particular vendor's technologies and guarantee that the hypermedia applications will continue to work without change as the underlying technologies evolve.

4.9. The Extensibility Rule:

A hypermedia system should be easily extensible and contractible in functionality and user interface in order to solve wide classes of application problems and to accommodate users of all types.

4.10. The Multilingual Rule:

A hypermedia system should support user interaction, user communication, and data storage and retrieval in languages other than American English language.

4.11. The Performance Support Rule:

A hypermedia system must accommodate both novices and power users, allowing them to author visually and to debug user-friendly applications that can become self-contained electronic performance support systems for their users. A hypermedia system must also be its own electronic performance support system.

4.12. The Standards Rule:

A hypermedia system should support and comply with all relevant formal and market standards. Standards reduce the number of choices to a tractable number. In so doing, they allow product vendors to innovate and add truly useful features to their products instead of devoting time to proprietary implementations of common functionality. And standards-adherence by vendors gives the consumers of their products the confidence that the products will be compatible with other products in those areass.

There are two types of standards, namely, 1) formal standards and 2) market standards. The former are debated and agreed upon by formal standards organizations or industry consortia.

For example, the International Standards Organization (ISO) over the years has published standards in different areas. ISO has defined-8-bit encodings for Roman character sets, the architecture and functionality of network protocol stacks, and the syntax and semantics of the SQL, databse - access language.

Market standards undergo a different process. They become standards by virtue

9 - Extensibility Rule,

10-Multilingual Rule,

11-Performance Support Rule, and

12-Standards Rule.

4.1. Multimedia Rule:

A hypermedia system should support both user-driven and time-driven input and output of the full range of sensory-rich multimedia types, including not only those items that are stored internally in the system but also those externally in other data sources.

4.2. The Object Rule:

A hypermedia system should employ object metaphors for system interaction, system storage, and application creation.

4.3. The Scripting Rule:

A hypermedia system should provide a rich, user-accessible scripting language for extending and modifying the behaviour of the system and its application elements.

4.4. The Multi-User Rule:

A hypermedia system should support the collaborative building of applications by multiple concurrent authors on networks of heterogeneous computers and the execution of those applications by multiple concurrent users.

4.5. The Scalability Rule:

Applications developed with a hypermedia system should continue to work well and with predictable performance characteristics when deployed in production environments that contain much more data and many more concurrent users than existed in the prototype or pilot version of the application.

4.6. The Interoperability Rule:

. Hypermedia applications should be able to exchange both data and control not only among themselves, but also with external applications and data stores such as SQL relational data bases.

4.7. The Hyperlink / Hyperview Rule:

A hypermedia system should allow users to establish different media types, to browse and navigate those relationships in an ad-hoc-linear manner, to determine

readily one's location among them at any time, and to return easily to any prior location or a navigated path.

4.8. The Technology Independence Rule:

A hypermedia system should provide independence from any particular vendor's technologies and guarantee that the hypermedia applications will continue to work without change as the underlying technologies evolve.

4.9. The Extensibility Rule:

A hypermedia system should be easily extensible and contractible in functionality and user interface in order to solve wide classes of application problems and to accommodate users of all types.

4.10. The Multilingual Rule:

A hypermedia system should support user interaction, user communication, and data storage and retrieval in languages other than American English language.

4.11. The Performance Support Rule:

A hypermedia system must accommodate both novices and power users, allowing them to author visually and to debug user-friendly applications that can become self-contained electronic performance support systems for their users. A hypermedia system must also be its own electronic performance support system.

4.12. The Standards Rule:

A hypermedia system should support and comply with all relevant formal and market standards. Standards reduce the number of choices to a tractable number. In so doing, they allow product vendors to innovate and add truly useful features to their products instead of devoting time to proprietary implementations of common functionality. And standards-adherence by vendors gives the consumers of their products the confidence that the products will be compatible with other products in those areass.

There are two types of standards, namely, 1) formal standards and 2) market standards. The former are debated and agreed upon by formal standards organizations or industry consortia.

For example, the International Standards Organization (ISO) over the years has published standards in different areas. ISO has defined-8-bit encodings for Roman character sets, the architecture and functionality of network protocol stacks, and the syntax and semantics of the SQL, databse - access language.

Market standards undergo a different process. They become standards by virtue

9 - Extensibility Rule,

10-Multilingual Rule,

11-Performance Support Rule, and

12-Standards Rule.

4.1. Multimedia Rule:

A hypermedia system should support both user-driven and time-driven input and output of the full range of sensory-rich multimedia types, including not only those items that are stored internally in the system but also those externally in other data sources.

4.2. The Object Rule:

A hypermedia system should employ object metaphors for system interaction, system storage, and application creation.

4.3. The Scripting Rule:

A hypermedia system should provide a rich, user-accessible scripting language for extending and modifying the behaviour of the system and its application elements.

4.4. The Multi-User Rule:

A hypermedia system should support the collaborative building of applications by multiple concurrent authors on networks of heterogeneous computers and the execution of those applications by multiple concurrent users.

4.5. The Scalability Rule:

Applications developed with a hypermedia system should continue to work well and with predictable performance characteristics when deployed in production environments that contain much more data and many more concurrent users than existed in the prototype or pilot version of the application.

4.6. The Interoperability Rule:

. Hypermedia applications should be able to exchange both data and control not only among themselves, but also with external applications and data stores such as SQL relational data bases.

4.7. The Hyperlink / Hyperview Rule:

A hypermedia system should allow users to establish different media types, to browse and navigate those relationships in an ad-hoc-linear manner, to determine

The list of MM types should be openended since the computer field advances dramatically whenever new media types are introduced [think of 1/0 devices such as the laser printer which drove the development of desktop publishing].

6- A hypermedia system is defined as a comprehensive MM system that also has hyperlinks between its "pages".

There are other definitions for the hypertext system, multimedia system and the hypermedia system as follows:

- A hypertext system is mainly determined through non-linear links of information. Pointers connect the nodes. The data of different nodes can be represented with one or several media types. In a pure text system, only text parts are connected. Thus, hypermedia is understood as an information object which includes links to several media.

- A MM system contains information which is coded at least in a continuous and discrete medium. For example, if only links to text data are present, then this is not a MM system. It is a hypertext.

A video conference, with simultaneous transmission of text and graphics, generated by a document processing program, is a MM application. Although it dose not have any relation to hypertext and hypermedia.

- A hypermedia system includes the non-linear information links of hypertext systems and the continuous and discrete media of MM systems. Fig.(2) emphasizes the relation among MM, hypertext (HT) and hypermedia (HM).

4 - ESSENTIAL ELEMENTS OF HM

There are twelve rules or criteria by which one can compare and evaluate hypermedia systems. These rules can be classified as follows:

1 - Multimedia Rule,

2 - Object Rule,

3 - Scripting Rule,

4 - Multiuser Rule,

5 - Scalability Rule,

6 - Interoperability Rule,

7 - Hyperlink / Hyperview Rule,

8 - Technology Independence Rule,

Application	Media Profile						Network Requirements		
	voice	data	text	image	audio	video	distributive	interactive	bandwidth
banking	*	*	*	*			H	M	M
insurance	*	*	*	*			H	M	M
medical imaging		*	*	*			L	M	H
real estate		*	*	*	*	*	H	H	M
education	*	*	*	*	*	*	M	M	M
distance learning	*	*	*	*	*	*	H	H	H
advertsing			*	*	*	*	H	H	H
publishing				*	*	*	H	H	H
travel agency				*	*	*	H	H	M
co-operative working	*	*	*	*	*	*	L	H	H
library			*	*	*	*	H	M	M
sales			*	*	*	*	H	H	M
training			*	*	*	*	H	M	M
medicine	*	*	*	*	*	*	M	M	M
hypermedia	*	*	*	*	*	*	H	H	H

Table (1)

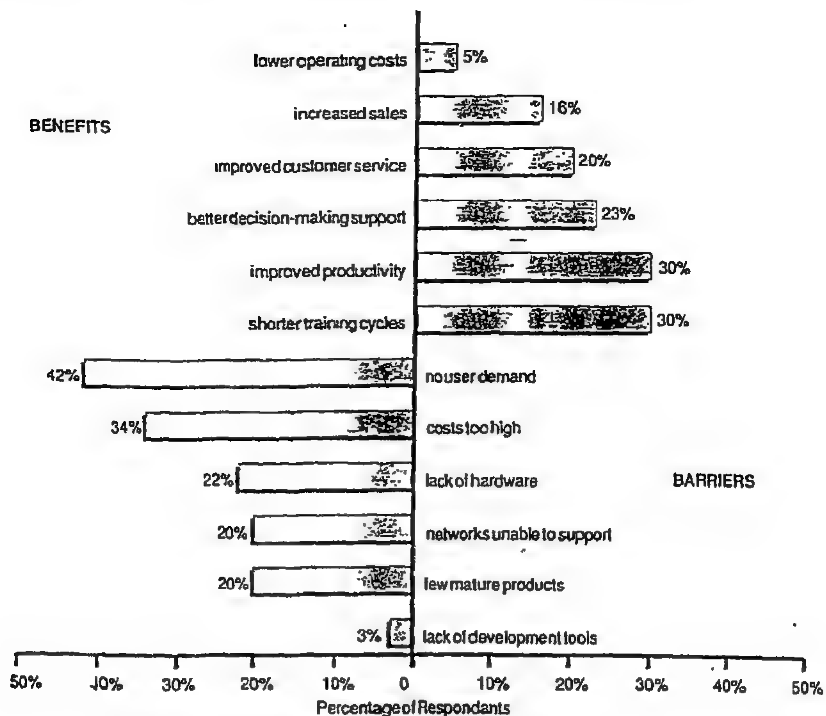
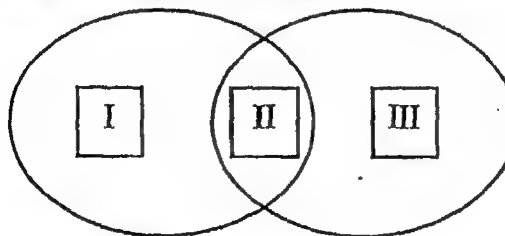


Figure 1 : Benefits & Barriers of Multimedia Applications.

5- A Multimedia system is a system as above where the pages may be of different individual media types or compound pages may include many media types.

There are many MM applications such as:
Banking, insurance,... etc, refer to table (1). The benefits and barriers of MM applications are shown in Fig.(1).



I = Multimedia (MM).
II = Hypermedia (HM).
III = Hypertext (HT).

Fig. (2): the relation among MM, HT and HM.

The decade of the **1990s** was bringing larger systems which work on multiple platforms, serve scores of users, and were becoming commercially significant.

Companies like MacroMedia and Gain Technology have comprehensive sets of products and are moving in different directions.

There are also important applications of these technologies in SW engineering. An early application was Neptune, and a current example is Ensemble, an environment for the uniform analysis and synthesis of SW and multimedia (MM) documents. It is a framework for the integrated support of interactive development of complex natural language and formal language document. The system has direct manipulation capabilities as well as services for formal description of syntax, semantics, and transformations of structured objects and representations. Users can edit and view compound documents of many types of components represented in a variety of media. Views can display the logical structure of a document as in a tree-structured representation or a view might display the output of a transformational process such as a printed view of a program.

3 - DEFINITIONS

1- The computer systems (CS) are operating by visiting a sequence of "pages", sometimes called cards or scenes.

2- A "simple page" might be a page containing text, graphics, animations and other objects.

3- A "page" would be made up of some number of simple pages. In a system that works with programs, e.g., Ensemble, a "page" might represent a "module" in the programming lauguage.

4- A hypertext system is a system as described above for which a user may also:

1- Create or delete pages,

2- Insert or delete one or more lines between two previously defined pages.

3- Edit pages,

4- Visit a page by following a link.

Thus, it is fair to talk about traversing a graph of pages in a hypertext document.

signers should meet these criteria and extend them. Consequenty, the technology will go forward and every body will benefit.

2 - HISTORICAL OVERVIEW

Almost (53) years ago, Vannevar Bosh wrote an article (Bush, 1945) in which a system was described.

This system was called **Memex** which could be said to have multimedia capabilities (typed items, photographs, and have written annotations on microfilm). The system also had associative memory and links representing "Trails" between its objects. It is fair to credit Bush with being the parent of these concepts even though the technology did not exist to implement them at time.

In the 1960s [Englebart, 1984, Englebart and English, 1968], the Augment System of Douglas Englebart had appeared. This system employed a mouse and an optional one-handed board keyset, and offered genuine hypertext capability. It was so much harder that the first system (which employed multiple media) would be identified, since the list of candidate systems is unclear. Ted Nelson's first saw that these principles could be extended to a hypertext network of all of society's documents [Nelson, 1981]. Moreover, he expressed his visions in colourful language ("If computers are the wave of the future, displays are surfboards" (Van Dam, 1988)

In 1987, the Hypercard program [Goodman, 1990] was supplied to all purchasers of Macintosh computers. While Hypercard may be regarded as a primitive hypermedia system, its widespread availability stimulated, interest in this area, and thousands of Hypercard stacks were created.

Notecards was developed at xerox from a different conceptual basis. It was intended as a supporting system for collecting and organizing ideas, each repressented by a card and organized by links.

Intermedia was a major multi-user hypertext system intended to focus on scholarly and educational applications. With its annotation service, link browser and linguistic tools, it intended to be both an authoring and reading tool. Certain implementation decisions have prevented intermedia from being adopted more universally. These are documented in Haar et al [1992].

The extensive multimedia work done in Project Athena at MIT was contemporaneous with the development of Intermedia.

CHAPTER 11

HYPERAMDIA SOFTWARE DEVELOPMENT TECHNOLOGIES

Farahat F. Farahat & Mohamed M. Essas

ABSTRACT

Hypermedia refers to a set of technologies that deals with a new method of: 1) organizing and 2) supplying associations between different elements of multimedia information.

It can deliver new freedom to users in exploring large amounts of mutimedia information at their own pace, and according to their own interests. Hypermedia will lead to very high levels of user interaction and consequently very low lattency/high bandwidth (BW) within the communication channel.

This paper presents: 1) the fundamental properties, that a system have to drserve the name **"hypermedia system"**, and 2) the features and technologies of hypermedia in the light of application requirements.

1 - INTRODUCTION

An explosion of interest in multimedia, hypertext, and hypermedia systems and software (SW) have been seen in the last few years. In 1987, the first major conference on hypertext took place at the university of North Carolina. Since that time, large amount of literature has been created and different commercial systems are now becoming available. It is now time to speculate about what is needed to develop the promise of hypertext and hypermedia systems.

Hypermedia has the potential to revolutionize the development and delivery of applications as well as the sophistication and usability of those applications.

In this paper, the fundamental properties, that a system must have to deserve the name **"hypermedia system"**, will be presented. Industrial users of such systems should demand hypermedia systems that meet this set of requirements. System' de-

CHAPTER 10

Enrichment Self-Learning Programmes in the Curricula of Primay Schools Utilizing Multimedia *

Prof. Dr. Aida A. Abou Gharib

&

Dr. Shaban H.A. Ibrahim

ABSTRACT

In the light of the current growing scientific and information progress, it is difficult to get satisfied with the existing curricula to reach achievement levels to fulfil the competence of the pupils in the 21st ceutury civilizatin. Therefore, it is important for the pupils to practice self-learning techniques through enrichment activities accompanys or paraellel to the curricula utilizing multimedia programs which are administered by computers. This fact underlines the need to conduct this study.

The investigation was conducted within the Division of curricula research of the National Center for Educational Research and Development. The steps are as follows:

1- The formulation of a team which works and assigns tasks according to specialization and courses in the primary schools, propose work plans, set up time schedule, for deliveries, and propose program criteria.

2- Design and production of the enrichment programmes for school curricula (Arabic language, mathematics, science... etc.) in Arabic lauguage for Arabic schools and language schools.

3- Implementing the proposed enrichment programmes for all stages of the primary schools on CD-ROMs in the Technological Development Center of the Ministry of Education.

4- Evaluating the proposed curricula and judging their scientific and technical corretness as well as their suitabllity fa primany schools pupils, besides the extent of achieving their obiectives.

* The original text is written in Arabic.

PART FOUR

MULTIMEDIA SOFTWARE, SELF-LEARNING PROGRAMMES AND EXPERT SYSTEMS

REFERENCES

1. R. Steinmetz and K: Nahrstedt, "Multimedia: Computing, Communications and Applications", Prentice Hall, 1995.
2. A.Y. Bilal, M. Zaki, and et al., "The Use of Multimedia Tools in Education", Workshop on "Multimedia in Eng. Edu." Cairo, May 1995.
3. Craig Partridge, "Protocols for High-Speed Networks: Some Questions and a Few Answers", Computer Networks and ISDN Systems, 25 (1993) 1019-1028.
4. Simon J. Gibbs and Dionysis C. Tsichritzis, "Multimedia Programming", Addison-Wesley, 1995.
5. Alaa Eldin Mohamed El-Ghazali, "Interactive Multimedia Development in the Arab World," in Towards the Development of Arabic Electronic Information Resources to Meet the Challenge of Civilization"Cairo, Academic Bookshop, 1997, pp. 181-217.
6. European Commission: "Directorate General XIII", ACTS: Advanced Communication Technologies and Services, 1994, pp. 51-54.
7. F. Bartsch and E. Auer, "Lessons Learned from Multimedia Field in Germany", IEEE Communications Magazine, Oct. 1997.
8. Digital, Networked Computing Catalogue, London; Digital Equipment Corporation, 1997.
9. Compaq Netelligent, Giga Ethernet Technology, Compaq Computer Corportion Networking Products Division, (November 1996.)
10. Csaba Szabo, "An Ethernet Compatible Protocol to Support Real Time Traffic and Multimedia Applications", Computer Networks and ISDN Systems, 29, (1997) 335-342.
11. Manfred Wittig, "Large-Capacity Multimedia. Satellite Systems", IEEE Communication Magazine, (July 1997).
12. Ra'ed Y. Awdeh and H.T. Mouftah, "Survey of ATM Switch Architectures", Computer Networks and ISDN Systems 27 (1995) 1567-1613.
13. William Stallings, "Local & Metropolitan Area Networks", Prentice Hall, 1997.

The exponential growth of Internet use may result in reduction of grade of service of telecommunication networks. Capacity increase is the challenge to telecom operators. Communication satellites "see" a large customer population and provide a large amount of - mainly entertainment - information. Interactive services are possible with enhanced communication satellite systems. [11]

7. CONCLUSION

Multimedia applications require Gigabit network. Multimedia networks have these features:

1. Bandwidth of Gbps or more.
2. Multicasting.
3. Real time constraints.
4. Reliability.
5. Quality of services.

It is obvious from the applicable study, which was performed at the GITEX '97 exhibition in Dubai that the greatest network companies have many solutions for multimedia networks. These solutions agree with multimedia network features mentioned above.

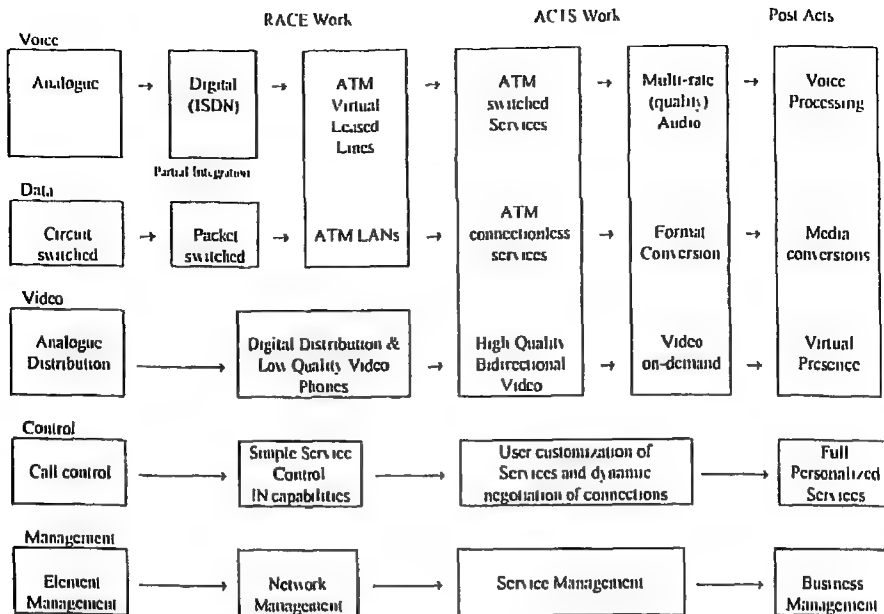


Fig. 5: High performance Networking - Technology Evolution [6]

Gigabit Ethernet is a logical extension of today's Ethernet and Fast Ethernet standards that will provide maximum performance for the client/server networks of tomorrow.

Gigabit Ethernet products are expected to be available starting in July, 1997, and the Gigabit Ethernet (IEEE 802.3z) standard is scheduled to be completed by January, 1998.

Local area network systems, in general, have been traditionally viewed as being incapable of carrying multimedia and/or high speed real time traffic due to data rate and protocol limitations. However, due to recent advances in video coding, on one hand, and the emerging high speed Ethernet versions, on the other data rate demands and resources seem to match quite well for many practical scenarios. Therefore, appropriate "integrated" media access protocols are needed that (i) support several classes of real time traffic; and (ii) compatible with the original data-only MAC protocol so that on an integrated/multimedia services Ethernet LAN, stations with integrated functionality can coexists with standard data-only stations. [10]

port system embracing different networking technologies and providing a platform for advanced multimedia communication services. In this sense, high performance networking must concentrate on quality of service in integrated networks, on overall signaling and management issues and related operational recommendations, and on robust network and transport protocols. Operational experience will help deduce organizational and technical constraints and define necessary agreements.

Realization of high performance networking will be in steps of evolution while keeping a backwards compatibility to "classical" data communication services and the connectivity and interoperability to the end systems (telephones, videoconference centers) of today's mass market networks. It goes beyond conventional data communication, as defined by the ISO-framework on "open Systems Interconnection", and beyond already established "audio/video-communications".

A realistic vision of evolution requires the observation of three different levels: [6]

- The development and introduction of high performance network (bearer) service with ATM as a starting point and end-to-end network/transport protocols, in order to ensure interconnection with established in-house and wide area networking technologies, which are capable of handling the different demands of multimedia communication.
- The development, introduction and management of high performance teleservices, by providing a service platform and thus ensuring interoperation between multimedia end systems.
- The introduction of "World Applications" linking users sharing the generic applications.

High Speed networking must include the two "lower" levels but with a clear view to and feedback from the application level.

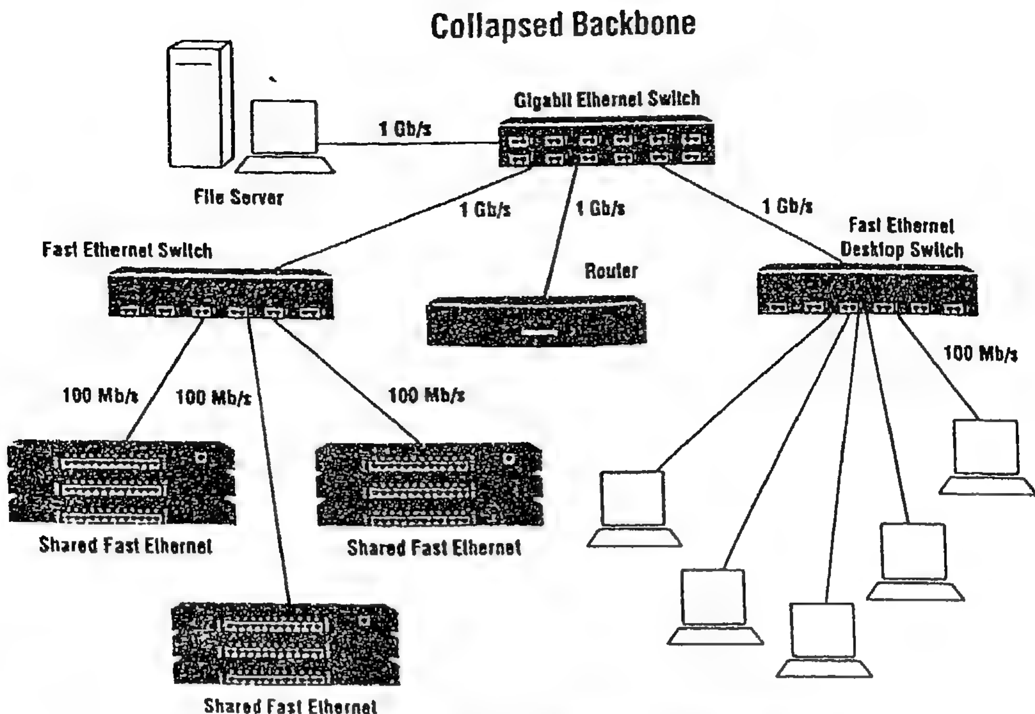


Fig.4: Gigabit Ethernet Application Example in a Collapsed Backbone [9]

6.THE LATEST AND THE MOST PROMISING TECHNOLOGIES IN THE NEAR FUTURE

The telecommunications field is now at a turning point where very high transmission speeds are complemented by an increased demand for this high speed. At the same time a cost-effective evolution from the current networks is required.

The planning of an ATM backbone network must be achieved with minimal disruption to existing network users while achieving economic benefits as soon as possible. A major issue for the future is to ensure the interconnection of advanced services on customer premises, corporate and public networks.

High development and investment costs will limit ATM based IBC coverage in the short to medium term. It is likely that existing networks will be extended progressively. Overlay networks will be provided with optical fibre access and later be incorporated within Integrated Broadband Communication - IBC.

To meet the objectives of this area, it is necessary to develop a broadband trans-

- Preserve the minimum and maximum frame size of the current IEEE 802.3 standard.
- Provide full-and half-duplex operation.
- Support star-wired topologies.
- Utilize the CSMA/CD access method with support for at least one repeater/ collision domain.
- Utilize ANSI Fiber Channel's FC-1 and FC-0 as a basis for work.
- Provide a family of physical layer specifications that support a link distance of:
  - At least 500 meters on multimode fiber.
  - At least 25 meters on copper (100 meters preferred).
  - At least 3,000 meters on single mode fiber.
- Support a maximum collision domain diameter of 200 meters.
- Support media selected from ISO/IEC 11801.
- Adopt flow control based on the IEEE 802.3x standard.
- Specify an optional GMII (Gigabit Ethernet Media Independent Interface).

Applications and Topologies :

Just as in 10 and 100 Mb/s Ethernet, Gigabit Ethernet will be switched, routed, and shared. All of today's Internet working technologies, as well as emerging technologies such as IP-specific switching and Layer 3 switching, are full compatible with Gigabit Ethernet. (The emergence of new protocols, such as RSVP, 802.1Q, and/or 802.1p, will provide multimedia support as well as VLAN and explicit priority support for Gigabit and 100 Base-T Ethernet). The initial application for Gigabit Ethernet will be in collapsed backbones between Fast Ethernet switches. repeaters, routers, and servers.

5.4 COMPAQ Applications :

The increasing complexity of desktop computing and bandwidth-hungry applications such as multimedia, medical imaging, CAD/CAM, and pre-press processing are demanding unprecedented LAN speeds. Out of all the availble high-speed LAN technologies, Fast Ethernet, based on the popular 10 Base-T Ethernet, has become the leading choice in meeting those demands by offering greater bandwidth and improved client/server response times. Now, however, the increased use of Fast Ethernet connections at the server and desktop is creating a need for even higher-speed network technology at the backbone.

An IEEE (802.3z) task force is developing a new Ethernet standard called "Gigabit Ethernet" that runs at 1000 Mb/s. Gigabit Ethernet employs the same CSMA/CD protocol, frame format, and frame size as its predecessors, plus interoperability and backward compatibility with the installed Ethernet. Gigabit Ethernet will also support existing applications, Network Operating Systems (NOS), and network management.

Gigabit Ethernet will initially be deployed as a backbone interconnection between 10/100Base-T switches and as a connection to high-performance servers.

In November 1995 [9], Compaq Computer Corporation recognized the need for Gigabit Ethernet technology and proposed the basis architecture to the IEEE 802 committee. In early 1996, the IEEE formed the Gigabit Ethernet Task Force (802.3z) whose commission was to draft a Gigabit Ethernet standard. Compaq has contributed significant technical input and resources to expedite the development of this standard.

In April, 1996, the Gigabit Ethernet Alliance was formed by Compaq Computer Corporation and a few other leading networking and computer companies to promote the Gigabit Ethernet technology and to ensure multi-vendor interoperability. Since then the Alliance membership has grown to more than 72 companies.

Gigabit Ethernet Characteristics :

Agoal of the Gigabit Ethernet Task Force is that the Gigabit Ethernet standard which meets the following criteria: [9]

- Run at 1000 Mb/s at the MAC/PLS service interface.
- Adopt the IEEE 802.3 Ethernet frame format.
- Meet the IEEE 802 functional requirement.
- Provide simple forwarding between 10 Mb/s, 100 Mb/s, and 1000 Mb/s Ethernet.

comfortably and seamlessly as your business evolves. Whether you need to upgrade next year, our switching solutions work with you in mind. DIGITAL offers a variety of high-performance solutions that protect your current investment while meeting you ever-expanding performance needs. Both the GIGA switch/FDDI and ATM systems have been recognized by the industry as top choices.

The GIGA switch/FDDI System - Named Data Communications "Hot Product of the Year" and the winner of R&D magazine's "R&D 100" award for technology leadership, the GIGA switch/FDDI system provides an easy way to implement switched networks today, with the flexibility to grow to ATM technology as you require.

The GIGA switch/ATM System - This high-performance networking switch has been designated as Network World's ATM Buyer's Guide's "Editor's Choice". The GIGA switch/ATM System, a 10.4-Gb/s ATM switch, is the only ATM switch that ensures network stability with FLOW master flow control, a zero cell loss traffic management technique.

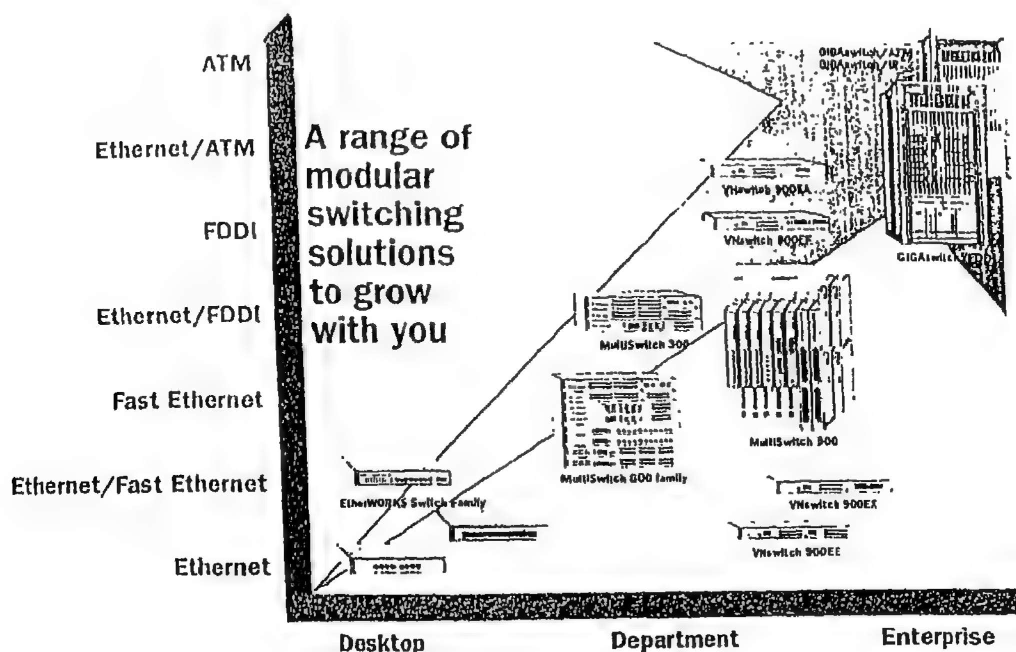


Fig. 3 : Digital Switches [8]

By combining the new GIGA switch/Ethernet system with Digital's 64 -bit Alphaservers, GIGA switches. MultiSwitches, VNswitches, Ether WORKS adapters, Internet/Intranet software, and web-based management, Digital offers the highest perfoming and most scaleable Gigabit/Fast Ethemet solution in the industry.

LAS VEGAS, May 6, 1997 [8]- At the Net World + Interop exhibition here today. Digital Equipment Corporation announced plans to extend its industryleading GIGA switch and MultiSwitch 900 enterprise switching platforms to support high performance Gigabit Ethernet solutions. Digital will offer the industry's broadest range of high-performance switching options including 10 Mbps, 100 Mbps, and 1000 Mbps Ethernet, FDDI, ATM, and IP switching for both cells and packets. Digital's proven, integrated GIGA switch and MultiSwitch architectures provide customers reliable, cost-effective migration options to support even the most demanding Intranet bandwidth requirements.

For proven ATM performance with industry-leading scaleability and flexibility, nothing beats the DIGITAL GIGA switch/ATM family. These ATM Forum-compliant systems deliver awesome throughput with guaranteed zero cell loss. And with ATM, IP Switching, and LAN emulation all on the same switching system, you get outstanding flexibility for adapting to changing network requirements without changing your hardware.

From small high-performance workgroups to large campus backbones, the GIGA switch/ATM systems scale seamlessly to handle your most demanding LAN, VLAN, and WAN applications while protecting investments. You get a choice of GIGA switch/ATM systems, each offering industry-leading performance to support a wide range of high-performance applications. The 5-slot modular chassis features the lowest cost for an entry-level system, supporting from one to 16 ports, while the 14-slot chassis offers high capacity-supporting up to 52 ports. Both systems provide component redundancy for large, high-volume backbone applications and use a nonblocking crossbar switch architecture for high throughput and low latency.

The DIGITAL family of switched networking solutions lets you evolve easily into switching technology without disrupting your existing networking infrastructure. Because our high-performance switching products coexist with existing LAN technology, your ATM and other switched networks can integrate with today's FDDI, Ethernet, and Fast Ethernet LANs. This means you add switched networking as needed to protect your investment in existing technologies, and later evolve those networks to full switch-based capabilities.

Plus, all our products are scaleable, no matter what size network you currently operate. DIGITAL has the technology in place so that you can start small and grow

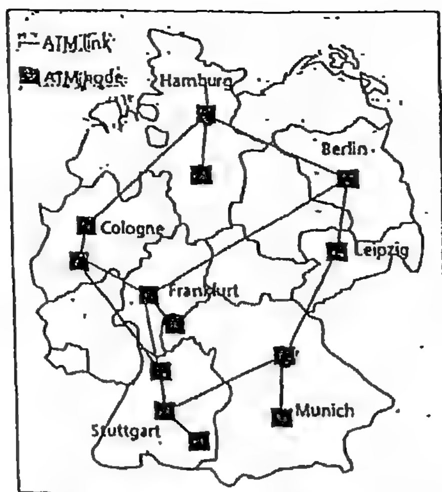


Fig. 2 : ATM Pilot Network [7]

5.3 DEC Applications :

ATLANTA, GA., October 8, 1997 [8]- Digital Equipment Corporation today announced its next generation Ethernet switch, the GIGA switch/Ethernet, providing up to three times the backbone capacity of competitive offerings. Featuring smooth migration from high-density Fast Ethernet to powerful Gigabit Ethernet switching, DIGITAL's GIGA switch/Ethernet enables customers to upgrade their networks to speeds of 1000 Mb/s as LAN bandwidth requirements of Intranets grow.

Digital has added a new high-density Gigabit/Fast Ethernet solution called GIGA switch/Ethernet to the GIGA switch family. When fully saturated, this intelligent 100/1000 Mb/s Ethernet switch delivers switching throughput of more than 33 million packets per second on a 45 + gigabit per second (Gb/s) backbone. Based on the same robust and reliable features as the GIGA switch/FDDI, ATM, and IP products, the GIGA switch/Ethernet switch offers up to 24 full-duplex Gigabit Ethernet ports, up to 120 auto-negotiating 10/100 Fast Ethernet ports, up to 60 fiber optic Fast Ethernet ports, or a flexible combination of these configurations. It is designed with a fault tolerant, non-blocking crossbar switch architectur with no single point of failure. And it is easily managed with integrated clear VISN Web-based management.

The GIGA switch/Ethernet is a cost-effective backbone solution that is complementary to existing 10 Mb/s and 100 Mb/s Ethernet technology. It shares the same advanced features as its GIGA switch siblings, including non-blocking crossbar technology, high throughput, scaleability, high availabilty and reliability, and quality of service. The GIGA switch/Ethernet can be integrated with other GIGA switches as well as with other members of the MultiSwitch 900 and 600 families.

that pilot ATM platform in preparation for public ATM services.

In addition, multimedia trials were set up focusing on services for residential customers in order to gain experience with multimedia applications specific to private households. Alternative technologies for residential broadband access needed to be evaluated in conjunction with multimedia application scenarios for residential customers.

Several infrastructures were employed to conduct the field trials. Business multimedia projects mainly utilize the ATM. The network platform was extended beyond the German border by linking with the European countries. For residential projects based on current telephone or cable TV infrastructures, different broadband access technologies were employed.

Switches of the ATM network are located in major German cities (Fig. 2) User-network interfaces (UNI) of 2, 34, or 155 Mb/s bandwidth are offered to interconnect multimedia applications on the ATM level, or terminal adapters to access legacy services. Permanent virtual and connections (PVPs and PVCs) are set up via a management system. Switched virtual connections (SVCs) are also available. The ATM network also forms the backbone for the 34-155 Mb/s research network connecting universities and research facilities.

Management of ATM connections is conducted from a central ATM network management center (ATM-NMC). A network management software was developed to support configuration and fault management of PVP connections on the network management layer. The ATM-NMC communicates with ATM switches via a Q3 interface based on the European Telecommunications Standardization Institute (ETSI) NA5 model.

Since the German ATM network forms part of a European ATM pilot network, means to manage ATM network in a multi-operator environmeınt were also developed and tested. Linking of network management systems via a telecommunications management network (TMN) X interface on a cooperative basis (Xcoop) is needed as an importantant prerequisite to automatically establishing ATM paths across interconnected networks. Within the European Institute for Research and Strategic Studies in Telecommunications (Eurescom) program, an Xcoop field trial for ATM technology was conducted among several European project partners. The procedure was to agree on a common specification of the information model at the X interface for ATM configuration and fault management processes as well as supporting communications protocol stack. Each partner would then implement the specification in its own laboratory using different platforms. In consequential tests the implementations were validated against each other.

152

So far only the peak bit rate traffic is being offered or experimented with ATM backbone network. Realizing ATM networks with statistical gain for optimum use of the network would have to be conducted to generate stable standards and specifications on mutually agreed definitions of traffic parameters.

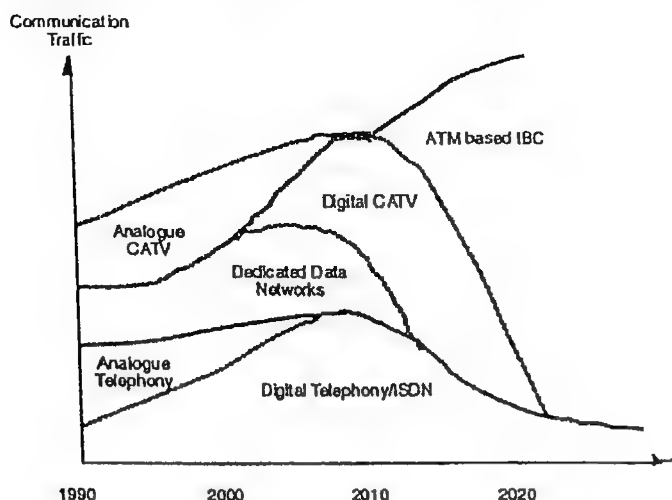


Fig. 1 : Evolution Timeframe [6]

The fibre optics infrastructure in Europe now amounts to more than 6.6 million fibre-km. This infrastructure offers almost unlimited bandwidth to permit a wide range of services to the user, such as HDTV, video on demand, multimedia, mobility, security, network management and others. The trend towards deploying ATM and laying more fibre is paralleled by the evolution on the application side. Organizations are now having a critical eye on their information processing capabilities and reassessing the way they transfer information with their customers and suppliers.

This technical evolution is mirrored by changes in competitive and regulatory environments. Network operators are starting to compete outside their territory. New entrants such as Cable TV operators, corportions with substantial assets in communication infrastructures (power utilities, railways, or others), teleport operators and mobile and satellite operators are also emerging. The blurring of the commpetitive lines means that a multiplicity of high-speed and lower-speed networks will coexist.

5.2 Germany Applications :

The experience gained with the Berlin metropolitan network led to building a German ATM pilot network in 1994, [7] which was later linked to the European ATM pilot network. A broad range of multimedia experiments was carried out on

Multimedia networks cannot totally ignore reliability since as errors increase, noise is added which ultimately leads to unacceptable presentation quality.

- **Quality-of Service (QOS):** Different applications have different communications requirements. A conferencing system, where data is presented once and tolerate a higher error rate than an application that records a multimedia data stream for future playback. Although less sensitive to errors, the conferencing system requires fast delivery, while for the recording application long transmission delays are of no concern. One way that networks can support variations in requirements is by allowing each application to specify 'Quality-of-Service' or QOS, parameters. For example, consider an audio stream where data is produced by a source at a constant rate of 64 kbps. An application might request the following QOS parameters: a bandwidth of 64 kbps, a maximum delay of 100 ms. and a maximum loss rate of 1%. The multimedia network would then allocate sufficient resources to satisfy the applications's demands, or tell the application that the network is 'busy'. [4]

5. APPLICABLE SOLUTION FOR MULTIMEDIA NETWORK

This section presents the applicable solutions for multimedia network.

5.1 Advanced Communications Technologies and Services (ACTS) Applications :

The high performance network area addresses the definition of the target networks, accounting for the emerging technologies, the identification of the missing element, the development of the means to control these networks, and the integration of all the pieces while verifying their operation through experimental usage. The technology of choice for integrated broadband communications is ATM, a successful outcome of the RACE programme. A major benefit of ATM is the unification of traffic types. The ATM technology acts as a multi-service integrator allowing to combine isochronous traffic (with real time constraints), connection oriented traffic, based on variable data rate, and connectionless service, over a single set of physical links. These characteristics are well suited for supporting multi-media services, which are already attracting all kinds of users and service providers.

ATM is also distsnce, protocol and speed independent. It can be implemented through the entire network, offering LAN-WAN integration and simplifying network management and operation. ATM cells can be sent over SDH transmission at speeds of 155 Mbit/s and higher, as well as over PDH at speeds of 34 Mbit/s or over LAN speeds of 100 Mbit/s .

3. WHAT DO APPLICATIONS REQUIRE OF GIGABIT NETWORKS

Some guarantees are made about the bandwidth available between the two computers. To see why this is so, consider an extreme case: 1 gigabyte of data is transferred over a network in which one link runs at only 19.6 kilobits; the transfer will take at least 115 hours.

Potential parameters for guarantees include:

- Network delay
- Bandwidth
- Reliability
- Failure recovery
- Service setup times
- Interarrival times

4. MULTIMEDIA NETWORKS FEATURES

Multimedia networks differ from current local and wide area networks in several ways. They are distinguished by the following features: [4]

- **Bandwidth:** Multimedia networks are likely to have bandwidth of Gbps or more; this is essential in the long-haul sections of the network where many simultaneous video streams can be expected. B-ISDN and ATM are examples of relevant technologies and the basis of international standards for future multimedia networks.
- **Multicasting:** Distributed multimedia applications often require multicasting, the transmission of data from one source to many destinations. The protocols for multimedia networks allow the set of destinations to change over time. This would be needed, for instance, in an electronic classroom where students can enter and leave as they choose.
- **Real-time constrants:** The transmission of multimedia data is subject to timing constraints include limits on transmission delay and limits on the 'jerkiness' of delivery. This is essential for applications that have 'live' sources or must present synchronized audio and video stream.
- **Reliability:** In multimedia networks, reliability is a question of a quality. Audio and video data, in comparison to text or numeric data, are less sensitive to errors.

and protocols for data transmission. [1]

Multimedia is the intergration of text, audio sound, graphic images, animations and full motion video. This technology is actually supported by three facts. [2.3.5.12]

1. The advent of VLSI that has led to producing powerful computers and work-stations with high performance and huge storage capacities.
2. The evolutions of communications technology towards broadband ISDN and ATM switching.
3. The advances in fiber-optics have produced high quality monomode fiber transmission systems capable of delivering gigabits of data per second over distances of hunbreds or thousands of kilometers with extremely small loss rates. Some of these fiber-optic systems are capable of transmitting terabits of bandwidth by using multiple gigabit channels.

This paper discusses multimedia networking needs, the new and applicable solutions as well as the features of multimedia network. This paper also presents a trip report about GITEX' 97 (Dubi) including interviews with its partners, such as DIGITAL, COMPAQ, FORE system, Microcom and 3 COM.

2. SOME KINDS OF APPLICATIONS NEED GIGABIT NETWORKS

This section presents a few number of possible uses of gigabit networks: [3]

2.1 Multimedia conferencing

One use is multimedia conferencing. Essentially, the idea that users should be able to participate in conferences in which users sit in their offices and communicate using voice, video and data connection to remote sites.

Multimedia Conferencing is expected to require gigabit bandwidths because the video communication consumes a lot of bandwidth. In a conference with several participating sites, the bandwidth requirements repidly approach a gigabit for video alone.

2.2 Virtual Reality (VR) :

Another interesting application is virtual reality. The idea is to simulate an environment so realistically that the user believes that he is in the new environment. To construct such an environment requires a computer to accurately mix sound, visual images and sensations such as touch and temperature. If there are multiple participants in a single simulation, several computers may have to exchange a lot of data to keep track of each participant's behavior in the common area.

CHAPTER 9

Multimedia Networking Technologies

Dr. Mohamed Mounir Essa

&

Dr. Alaa El Din Mohamed El-Ghazali

ABSTRACT

Desktop conferencing systems and other distributed applications require digital networks for the transfer of audio and video data. Many current networks are not appropriate for these applications, the protocols that govern current networks and their limited bandwidths are ill-suited for digital audio/video traffic.

Multimedia networks, on the other hand, provide a better match to the communication characterisitcs of distributed multimedia applications. These networks are specifically designed for multimedia traffic. Multimedia networks differ from current local and wide area networks in several ways. They are distinguished by different features, these features are presented and discussed in this paper.

Keywords: multimedia, gigabit networks, Asynchronous Transfer Mode (ATM) reference model, performance guarantees, Broadband Integrated Service Digital Networks (B-ISDN), ISO (OSI).

1. INTRODUCTION

Multimedia computing and communications are areas of intense current interest, software and hardware development, and future promise. Multimedia standards organizations are actively producing new standards for the field. Yet, the term "multimedia" and the subject areas it covers remain. In the future, all computers and networks will support multimedia computing and communication to provide appropriate services for multimedia applications. The high-speed networks with their higher bandwidth and transmission possibilities of all media kinds, have led to networked multimedia systems. [1]

A multimedia networking system allows for the data exchange of discrete and continuous media among computers. This communication requires proper services

REFRENCES

1- R. S. Pressman, "Software Engineering: A Practitioner's Approach", 3rd edition, McGraw - Hill Inc., New York, 1992.

2- J. A. McCall, and J.P. Cavano, 'A Framework for the Measurement of Software Quality", ACM Software Quality Assurance Workshop, Nov. 1978.

3- C. Jones, "Programming Productivity", McGraw Hill Inc, New York, 1986.

4- B. Beizer, "Software System Testing and Quality Assurance", Van Nostrand Reinhold, 1984.

5- P .J. Denying, "Computers Under Attack", Addison-Wesley, London, 1990.

6- J. D. Musa, A. F. Ackerman, "Quantifying Software Validation: When to Stop Testing?", IEEE Software, May 1989, pp. 1-3.

constraints of fully utilizing computer technology, the number cutting would range from 37%,53% and 57% of top, middle and supervisory levels respectively. Obviously, plans for switching training programs or other candidate plans should benefit from such kinds of results.

2- These estimates can present a value added measures of employing computer technology as reflected to each category of managerial levels. This would even has its own impacts when analyzing feasibility of projects moving some of the commonly listed tangible benefits into the domain of tangible ones.

3- It could be also considered as targeted figeted figures of enhancement of manager's time towards efficient achievements of their tasks. In other words, one would claim that almost a percentage over 50% saving of middle managers could be saved when properly trained to use successfully applied computer technology relevant to his domain of activities.

4- The estimates given here could be debated by different arguments including the relative weights suggested for each level as given in Table (1). However, these figures have been selected via correlating both reported measures by business experts [1], and those resulted from the empirical study which covered over 50 manager each level. More truthful measures may result for different application fields, however in analogous manner.

5- Debates could arise also from the suggested weighted impact values listed for each business task, however it's the intention of this research to enhance the study via considering the applicability of more adequate forecasting models ;properly the fuzzy logic.

To sum up, this paper has presented a model with a number of guidelines to evaluate empirically and analytically the impact of technology on different managerial levels. The importance of having such measures have been highlighted focusing on their great help to automating projects as well as re-engineering organizational structures. It has been the intention of this paper to highlight and to stimulate further research towards quantifying the problem of computer technology impact on managerial levels from different perspectives of work practices, job definition and number cutting if targeted. Even debates would stimulate others to enrich such kinds of problems which lack intensive research!.

EIF = RIF * TUF * UF where :

EIF: Enhanced Impact Factor
TUF: Technology Update Factor
UF: Utilization Factor of Technology
RIF: Relative impact factor

The Relative Impact Factor (RIF) of individual levels as given in Table (2) as: 0.37, 0.44, 0.57 for top, middle and supervisory levels respectively. The relative weight of the Technology Update (TU) is given to account for further improvements that may take place in terms of more problems that can be supported by computer technology due to the fast changeable characteristics of computer technology **First** the degree of successful Utilization of technology in the business mission of managers is considered by the factor UF in the equation. Such a factor can be estimated as the average percentage resultant from evaluating the degree of trend towards letting technology override the traditional work practices and the degree of successful implementation of technology resources.

Table (2) Estimate of the Weighted Impact Function of Computer To Main Job Functions Of Management

	Top Level		Middle Level		Supervisory	
Function	WIF	CIF	WIF	CIF	WIF	CIF
Planning	0.35	19.25	.5	10	.6	3
Organizing	0.25	2.5	.30	9.9	.5	2.5
Controlling	0.80	4	0.85	11.9	0.9	39.6
Analyzing/ follow-up	0.50	2.5	.6	9	.8	4.8
Communicat-ions	0.40	2.8	0.4	8.8	0.25	3.5
Decision Making	0.40	5.6	0.5	2	0.5	1.0
Supervising	0.20	0.8	0.2	2.5	0.2	4.0
Average Impact	37.45%		53.6%		56.9	

Second, the estimated impact of technology on individual levels can be looked upon in a number of different ways:

1- It could be employed as an indicator to those interested in number cutting when considering the reengineering of organizational structures of business organizations. For instance, under conditions of proper implementation and

* Computer technology supports managers at different extents basically in the form of: managing information, reporting, skills and requirements. For instance; if the weighted impact function is 30% of the problems encountered by the planning needed by a certain managerial level while the planning covers a 20% of the job span of needs; then the correlated impact factor (CIF) of technology on this attribute would be 6%.

5-Estimate the overall impact factor of technology for each managerial level by calculating the accumulated sum of the correlated impact functions over the span of business mission of each management level. This would come up with an overall measure of the impact function as corresponding to individual managerial levels. For instance, the estimated overall impact of technology on top management would be the accumulated sum of the individual CIF measures as defined from step - 4 for planning, organizing, control and the rest of the business attributes characterizing the business need for the top managerial level. Similarly, other measures can be estimated for the both middle and supervisory management levels.

RESULTS AND CONCLUSIONS

The research has intensively reviewed many of the recent reported literature covering advances and development in different computer fields. The proposed model has been applied to estimate the averge impact function of technology on each business mission. Table(2) summarizes the results obtained by correlating the relative impact figures with the weight factors of the main job requirements previously listed in Table (1).

In order to highlight the estimated results as given in this paper taking in considerations the main assumptions and rules suggested by the model, the following comments are summarized:

First, it is important, prior discussing the implications of the estimated results given in Table (2), to state here that those estimated figures are subjected to change due to a number of reasons in additions to the aforementioned ones in the basic assumptions of the model. These include, the dynamic feature of technology as well as the mentality trends of managerial levels towards letting technology overload their traditional work practices. In order to enhance the model, the resultant percentage of technology impact on each level can be further refined by considering subjective weight factors together with two more other parameters; one for the further improvement in technology and the other to account for successful implementation of CBIS as well as proper use of technology support to justify the different trends towards applying properly the technology. An Enhanced Impact Factor (EIF) can then be defined for each level of management as given in the following equation:

technologies. Table (1) summarizes the results of cited frequencies of the skills and needs for each category of the managerial levels. Obviously, different application fields may have different values, however this would serve as an evaluation methodology the cab be similarly applied.

Managerial level Skill/Need	Top Managers cited frequency	Middle Managers cited frequency	Supervisors cited frequency
planning	55%	20%	2%
Oranizing	10%	23%	4%
Controlling	5%	16%	44%
Analyzing/ Follow-up	5%	15%	6%
Communications	7%	12%	22%
Decision Making	14%	4%	2%
Supervising, monitoring, observing	4%	10%	20%

Table (1) Results of cited frequencies of the skills and needs.

The figures given in Table (1) has been based on imperical study which covered over150 managers from the selected business field of common characterisitics. On the other hand, the results whenever revealed great dispersion have been correlated a number of agreed-upon figures as suggested by the management models especially the one given by J. Kanter [2], which have proven quite acceptable matching. According to the proposed methodology, we have attempted to map these figures into relative needs from the major functions supported by computer-based-information systems.

Before proceeding into the analysis model, It is important to highlight the following comments:

* although different levels of managers have different job descriptions leading to some divergence in their Business mission, they all perform the listed needs/skills in Table (1).

1- Identify the main attributes of business mission of individual management levels by analyzing the jop characteristics in terms of the relevant tasks and skills needed to accomplish the job efficiently. Models suggested by business management's experts would be a starting point to be further refined via empirical as well as ststistical analysis to define a distribution of major tasks over a certain job span of needs and requirements. Analysis of such attributes should focus on categorizing the different skills and tasks from perspectives of their processing requirements. This would, in turn, result in a number of tasks and functions such as those given in Table (1) which accounts for the relevant business tasks identified for managerial levels.

2- Apply forecasting models to estimate status of relevant computer technology fields of more pronounced impact on business mission as identified by the previous step. Such a step should come up with figures characterizing the relative contribution of technology on major application categories such as planning, organizing, follow-up, decision making ..etc. In result, an accumulative figure indicating the direct impact of the technology as corresponding to individual job needs and requirements irrespective to the different managerial levels can be estimated. The estimated measure of technology impact on each considered function is defined as **'Impact Function' (IF).** For instance, an IF of 80% as resulted from forecasting analysis of the evolving technologies as corresponds to the planning tasks. Obviously, such a figure would reflect the increasing success in the problems encountered by artificial intelligence as well as dedicated EIS supporting planning type operations.

3- Define a weighted impact function(WIF) of technology for each level of management. Such a weighted impact function is derived by weighting the estimated relative impact of the step -2 to account for the different domains as well application definition of a certain business function which differ from one level of management to another. For instance, planning while been necessary to both top and middle management has different scopes of planning problems. Obviously, the top management would deal with higher degrees of unstructured problems rather than the case with other managerial levels. attributes of the business mission listed in Table (1).

4- Correlate between the weighted impact function (WIF) as result from step-3 with the relative weight of individual jop description as identified in the first step. **A correlated impact factor** (CIF) for each category of business mission can then be calcuated as the product of the weighted impact function (WIF) with the relative weights of jop percentage each level exerts or require to satisfy the specific listed job requirements. However, there has been quite an agreement in outlining the requirements from perspective of basic functions supported by the computer-based

ANALYSIS OF THE IMPACT OF COMPUTER TECHNOLOGY A PROPOSED MODEL

Developing a quantitative model to analyze the impact of computer technology has to face a number of challenging problems:

* There has been no specific criterion to quantitatively weight the effect of individual advances of the different computer fields to each level. For instance, improving transaction processing while has direct impact on the business missions of clerks also improve the flow of information to the managerial levels.

* Many of the listed figures about the evolving technologies over time have been based on logic judgment with intention to show the grow up of the targeted techniques over time. For instance, success reports of developing more dedicated Expert Systems, or maturity of developing some EIS to assist executive management are given to illustrate how relatively the technology has encountered more semi structured type problems to computer solutions.

* There has been; to the best of my knowledge; no correlation mechanism that would relate or weight the impact of certain evolving technique or attribute of the computer technology to the different levels of management.

The aforementioned factors reveal the challenging nature of the targeted model. However, it has been decided to suggest a guideline criterion which can be applied to find a correlation mechanism between the business mission model defined in the preceding preceding section and the different forms of computer technology attributes.

In order to explain the model, a number of important guidelines and assumptions need to be highlighted. **First,**We have to accept the fact that certain impact on a lower level of managament would indirectly leads to the improvement of work procedures of a higher level. However, there are some features which would appear of more impact on a certain level rather than on the other level. For instance, long term planning and organizing is more relevant to the top management rather than to the other levels. **Second**, there are a number of qualitative measures regarding the proper use, satisfaction, and successful implementation of the computer projects in the host organizations. **Third**, remains the fact that availability of computer based solutions does necessary means that technology resources are highly utilized to their maximum extent.

Consequently, in order to simplify the matter the model has considered the assumption of applying the available technology to full extent. Figure (3) Summarizes the analysis criterion adopted in this research; with the following steps:

ANALYSIS OF THE BUSINESS MISSION OF MANAGERS

The business mission of different managers has a number of common characteristics in terms of their processing needs. Most literatures have identified a number of atributes towards modeling the business mission of managers. In order to develop our own refinement criterion we have considered the following methodology towards setting a quantitative frame for the job processing needs as well as skills needed for each level of the managers from the perspectives of data and information processing:

1- define a detailed list of responsibilities and requirements based on the job description.

2- for each accountable responsibility, a number of skills to guarantee effective perfomance of the job responsibilities.

3- perform a statistical frequency analysis based on a satisfactory sample of managers 'job description. For instance, we compute the relative percentage of cited skills needed from the entire span of responsibilities identified for each manager.

4- Analyze the cited frequencies of targeted skills from perspectives of its main processing requirement: planning, organizing, decision making, managing data and/or information, controlling, administrating, and making directions as well as communications.

In order to highlight the aforementioned procedure, an in-depth analysis has been made to a number of published manager's careers based on proper job description. A questionnaire has been prepared and used to verify the adequacy as well as the realistic implementation of such listed needs and skills. The study has considered a set of companies working in the field of contacting, engineering and distribution covering a range from 20-30 managers from each level. The results have shown some basic differences in terms of the monitoring problems, supporting the decision making, automating work procedures and flow of data communication.

The listed needs of the manager's skills can be supported or categorized in the main functions of CBIS. For instance, controlling analyzing and monitoring are typical functions of MIS and Decision support systems.

Figure (2) summarizes the main functions supported by an CBIS as reflected to the different managerial levels. The relative presented functional areas in Figure (2) has intentionally attempted to give relative weighting of the impact of each function in terms of the business mission of managerial levels.

managerial work domain are highly unstructured, besides that successful ES 's cover a very specific scope of the work domain.

* Executive Information Systems, EIS while being introduced in the last few years, target improving the work mechanisms of executive managers as a special enhanced form of MIS and DSS. Shch systems need to face challenging nature of highly structured problems that managers need to solve.

* Office Automation has gained a dramatic jump in terms of their impact on the flow of data and information in and out organizations reducing much the time and paper work. Recent technology has contributed significantly to new forms of automation including electronic mail, conferencing and services through networks especially the INTERNET.

An in-depth study to the impact of technology on managerial level would then become important but difficult if one targets detailed undoubted figures in a quantitative manner. The difficulties of such analysis can be summarized due to the following:

1- Both technology and managerial business missions are highly dynmic due to the every day innovations and advances.

2- Study need to probe into the processing needs of the business such a study would mission of individual managerial levels

3- It would be important to retrieve logical indicative figures from the continuously changing power of computer technology in different fields as well as different functions of CBI's.

4- The main problem would be then, the correlation between the needs of a manager's business mission and the extent the computer technology would satisfy, a problem that has been always for debate!!.

The following sections would attempt to introduce analyze the main problem of this research along the highlights of the aforementioned difficulties.

number of logically accepted as well as interesting results from the perspectives of extent that technology can do as a replacement of human managers as well as the importance of reforming their work practices and roles. For instance, results has come up with figures ranging from 30% to 70% of dramatic change in the definition problems regarding both number and new work procedures of the business mission of different levels of managers.

INTRODUCTION

Computer technology has been dramatically improving the information systems in most fields of business applications. Many literature have addressed the scope as well as different examples of applying computer technology in many fields, even from managerial level perspectives [1, 2, 3]. While most specialists agree upon the importance of relying on such technology, there are a number of questioning regarding the extent that such techology would reduce the need to different managerial levels. Obviously, work procedures as well as management practices of the managers have to change in presence of successful Computer-Based-Information Systems (CBIS). Figure (1) presents a model indicating the different components of a CBIS from the Input-Process-Output perspectives. From the figure, managers are directly included in the two components of the knowledge workers and methods and procedures, while also implied by the different other components of the model. Meanwhile, most specialist have agreed upon classifying CBIS in terms of their functions into a number of types. Figure (2) illustrates the span of types as reflected to the different pyramidal managerial levels. In this figure, few comments would be important for subsequent analysis:

Management-Information-Systems (MIS), would provide managers with information in different forms such as summaries, statistics, graphs as well as highly structured modeled problems. Despite the value of such types in terms of improving managerial reporting as well as clear monitoring to the work problems, such types wouldn't help much directly in the decision making process!.

* Decision Support System complete the value of the preceding MIS to the extent of providing managers with useful indicators to solve problems, thus supporting the decision making process. Although, dialogues of What-If? Question experience answer sessions are still relying much on the knowledge as well as the quality of the still relying much on the knowledge as well as the quality of the managers.

* Expert systems while need to be field specific have employed advances in artificial intelligent and computers, attempt to provide managers with almost ready to go solutions. However, it is important to remind here that many problems in the

CHAPER 8

An Analytical Evaluation of the Impact of Computer Technology on Manageri-al Levels

Dr. Nashaat El-Khameesy

ABSTRACT

The impact of computer-based information technology has been addressed in many literature's, however in more or less in an abstract view. No one denies the importance of computers, however, very few have tried to analyze such an impact in quantitative manner. In most cases, information specialists have to face both declared and hidden forms of fractions while achieving their automation activities in organizations. It would be quite important to analyze different forms of the impact attempting some acceptable figures of the extent that such technology would impose on the managers from different attributes. Such studies have to consider some form of analytical as well as forecasting models to reach to indicators good enough to for specialists as well as organizations in terms of impact on the number of managers, their work practices and other related aspects. It would also highlight the truthfulness of fear from computers ending in labor number cutting as well as redistributing powers in a certain organization.

Throughout this paper, an analytical study has been made to quantitively estimate the effect of information technology on managers based on two main axioms: First, the nature of the business mission of individual managerial levels has been analyzed. Second, the extent of contribution of the computer technology; based on the undoubted figures of the new advances, has been correlated with the basic needs of the managerial levels. The study has also considered some typical comparative figures based on a statistical study on a group of 150 managers from fifteen Egyptian companies working in the sector of construction from the general business sector. Such figures have also been compared to similar results from international successful announced companies in literature. The study has resulted in a

velopment, supply and maintenance of software", International Organization for Standardization, 1991.

[14] International Standard ISO/IEC 9126; "Information Technology - Software Products Evaluation- Quality Characteristics and guidelines for their use", International Organization for standardization, Interanational Electrotechnical Commission, 1991.

[15] Working document for technical report of ISO/IEC JTCI/WG6; "Software Quality Evaluation Guide part 1: General Guide", by M. Azuma, Draft-Version 3, October 1992.

[16] I. J. Lloyd, and M. J. Simpson; "Fourth Report on the Legal Aspects of SCOPE"' SCOPE document SC 92/003/STH.il.ms/T.2.5/.DT/01, 25th February 1992.

[17] SCOPE Consortium, edited by H. L. Hausen, N. Cacutalua, and D. Welzel' "A Method for Software Evaluation and Certification - The Information Model - Refinement of the Phase One Model", SCOPE report SC. 92/016/GMD. HCW/T2.1.1.2./RP/02, GMD Sankt Augustin, Germany, February 1992.

REFERENCES

[1] B. W. Kernighan, and P. J. Plaugher; "The Elements of Programming Style", McGraw-Hill, N. Y. (1974).

[2] T. P. Bowen; "Specification of Software Quality Attributes", Rome Laboratory, New York, Technical Report, RADC-TR- 85-37. Vols. 1-3, 1976.

[3] B. W. Boehm, J. R. Brown, M. Lipow, G. J. Macleod, and M. J. Merritt; "Characteristics of Software Quality", North-Holland, Amesterdam, 1978.

[4] B. Kitchenham; "Towards a Constructive Quality Model", Soft. Eng. Journal, 105-112, July 1987.

[5] M. Deutsch, R. Willis; "Software Quality Engineering", Prentice-Hall, Englewood - Cliffs, NT: 1988.

[6] R. Nance, "Software Quality Indicators: An Holistic Approach to Measurement", Proc. 4th Annual Software Quality Workshop, Alexandria Bay, New York, August 1992.

[7] ISO/IEC Standard, ISO-9126 Software Product Evaluation - Quality Characteristics and Guidelines for their Use, 1991.

[8] M. C. Paulk, B. Curtis, M. Beth; "Capability Maturity Model for Software", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, August 1991.

[9] H. L. Housen, N. Cacutalua, D. A. Welzel; "Method of Software Evaluation and Certification- Draft", Arbeitspapiere der GMD, No. 571, GMD Sankt Augustin, September 1991.

[10] IEEE Std 610. 12-1990, "IEEE Standard Glossary of Software Engineering Terminology", Institute of Electrical and Electronics Engineers, Inc., NY, December 1990.

[11] ISO/IEC Guide 25; "General Requirements for the Competence of Calibration and Testing Laboratories", International Standards Organization, International Electrotechnical Commission, 1990.

[12] International Standard ISO/CD 8402-1; "Quality Concepts and Terminology Part One: Generic Terms and Definitions", International Organization for Standardization. December 1990.

[13] Interantional Standard ISO 9000-3; "Quality Management and Quality Assurance Standards-Part 3: Guidelines for the application of ISO 9001 to the de-

6. CONCLUSION

The paper provides a description of a software evaluation procedure. This evaluation procedure is a four-step procedure. The description is presented in terms of actions and results in order to cover the practical issues of planning, controlling and reporting on an evaluation.

The evaluation procedure proposed is adaptable to circumstances of any testing entity (available personnel, evaluation techniques and tools), and flexible to relevant standards of product type dependent quality requirements and of software engineering processes, and to legal or contract issues.

The evaluation framework proposed is designed to allow technology changes without needing modification of its basic principles. This is a consequence of the approach taken to use evaluation modules. Although, the evaluation framwork explicitly refers to ISO/IEC- 9126, other quality models can be applied without any serious change to the evaluation procedure.

Running a product evaluation provides information that can help to estimate the costs, improve evalution modules, or to identify needs for new evaluation techniques. The experience gained, therefore, should be stored in a database for the benefit of future evaluations.

One prime objective for developing this evaluation framework was to ensure that an evaluation process is effective. In order to achieve this goal, many issues have been considered, **first**; experience with evaluation modules and the module library, **second**; appropriateness of levels and software characteristics, **third**; appropriateness of product and process representation, **fourth**; calculation of actual costs in order to improve cost estimates, and appropriateness of the evaluation procedure.

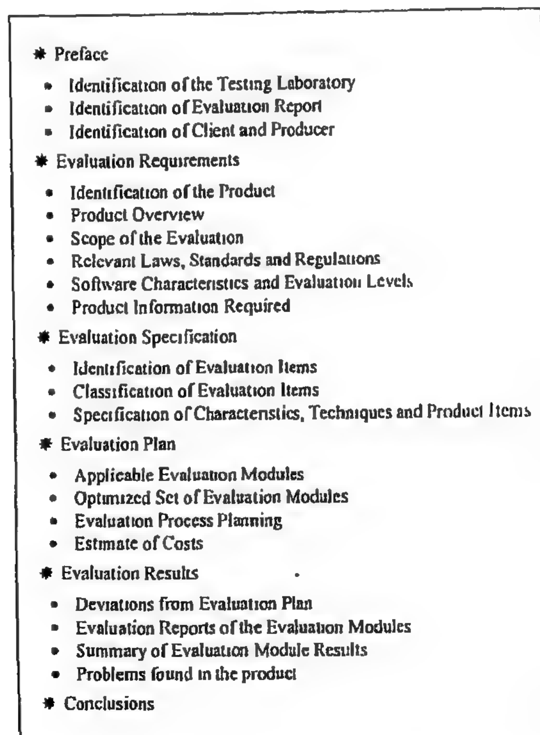
4.4. Reporting the Evaluation :

The last activity of the evaluation process is to collect all results, document the process and structure them in the evaluation report. The evaluation report compiles all outputs from the individual modules, including all observations made during the evaluation process, into the evaluation results, which are a part of the evaluation report.

The evaluation shall be reported accurately, clearly, unambiguously and objectively, as required by ISO/IEC Guide [11]. This includes all the information necessary for the interpretation of the evaluation results and all information required by the method used in order to be able to control application of evaluation procedure and the suitability of decisions taken.

5. STRUCTURE OF THE EVALUATION REPORT

A proposed structure of the evaluation report including its items is shown in Figure 9.



* Preface
  - Identification of the Testing Laboratory
  - Identification of Evaluation Report
  - Identification of Client and Producer
* Evaluation Requirements
  - Identification of the Product
  - Product Overview
  - Scope of the Evaluation
  - Relevant Laws, Standards and Regulations
  - Software Characteristics and Evaluation Levels
  - Product Information Required
* Evaluation Specification
  - Identification of Evaluation Items
  - Classification of Evaluation Items
  - Specification of Characteristics, Techniques and Product Items
* Evaluation Plan
  - Applicable Evaluation Modules
  - Optimized Set of Evaluation Modules
  - Evaluation Process Planning
  - Estimate of Costs
* Evaluation Results
  - Deviations from Evaluation Plan
  - Evaluation Reports of the Evaluation Modules
  - Summary of Evaluation Module Results
  - Problems found in the product
* Conclusions

Figure 9 : The Proposed Stucture of the Evaluation Report.

- Identifcation of a list of "metrics" applied (in case of checklists, it is the collection of questions applied).

- "Achievement inforamtion" i.e. coverage of checklists.

- Rating the results i.e. mapping onto the categories excellent, good, fair, poor (or others).

The actual results are to be compared with the required results specified in the evaluation specification. An evaluation conclusion shall be done according the pass/fail decision criteria- if they are defined. If the decision is pass the evaluation specifications and the actual resuls should be only repeated. If the decision is fail, the measurement result that drove this decision should be highlighted.

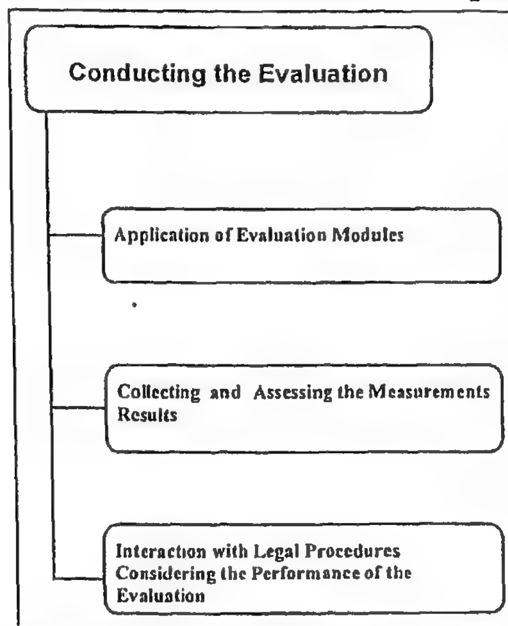


Figure 8 : The Actvities of Conducting the Evaluation.

In most cases, the evaluation may be expected to follow provisions and procedures laid down in the evaluation specification and plan. From the legal standpoint, the attempt must be made to identify and provide for potential problems. The most significant problem would be that the evaluation plan cannot be implemented. This may be caused by a number of factors of circumstances including the possibilities that :

- Deficiencies appear in the evaluation plan.
- The software may not be as specified.
- Third party intervention may prevent performance.

A first rough estimation of costs is obtained based on the evaluation levels. Here a refinement of the estimation based on the selection of evaluation modules is considered. Each module includes data from which the cost of its application can be derived. From this information the cost of implementing the evaluation plan is estimated. Costs might be divided into cost of personnel and cost of applying tools.

In the interaction with legal procedures concerning the evaluation plan, the client requests the production of an evaluation plan. This act constitutes the client's acceptance of the testing laboratory's offer to draw up the evaluation plan subject to the terms and conditions specified in the evaluation specification. The evaluation plan may restate many of the provisions of the evaluation specification but will also:

- Identify specific evaluation modules to be applied.
- Prescribe a timetable for the evaluation process.
- Provide for logistics of the evaluation process.

The testing laboratory may offer to conduct an evaluation according to the evaluation plan.

4.3. Conducting the Evaluation :

The implementation of the evaluation plan means applying the evaluation modules on the related product parts and collecting for each of them the application results. The output will be a collection of measurement reports resulting from the applicatition of the modules. The execution of the evaluation modules and the interpretation of the results migh be supported by a testing laboratory's workbench, assistant or advisor.

Measurements can be manual, computer aided (e.g., using a check list manager for applying check lists), or automatic (e. g. measuring the complexity in a source code component using a static analyser).

The main task is to collect the measurement result and also to keep any information (measurement data) about the measured product part that could be helpful for an acceptance decision to be taken. Figure 8, shows the activities to be conducted by the evaluation conduction.

Based on the results of evaluation modules, a synthesis of all measurement reports might be necessary. This can be done by one of the following two possibilities :

- Application of so called "Assessment Modules" where each of them defines a multidimensional assessment method.
- "Summarizing" output of all evaluation modules by :

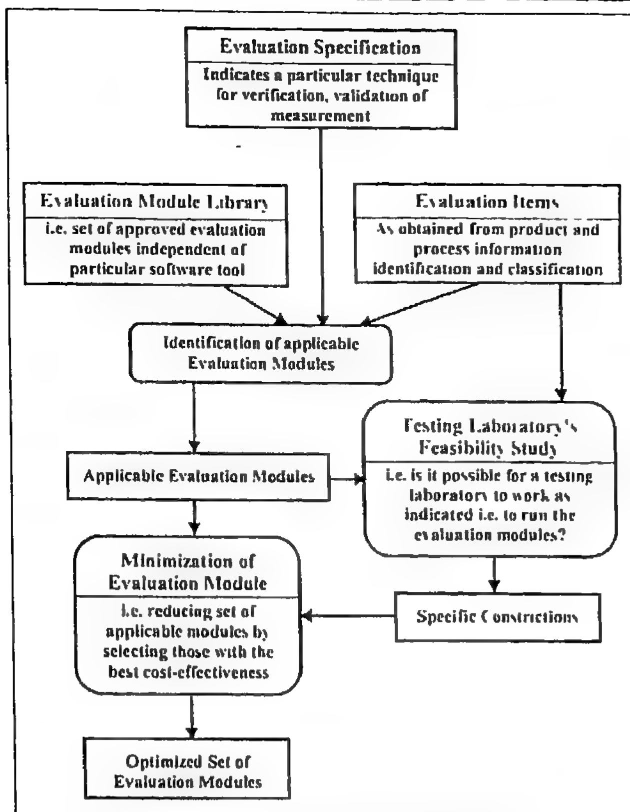


Figure 6 : Selection of Evaluation Modules.

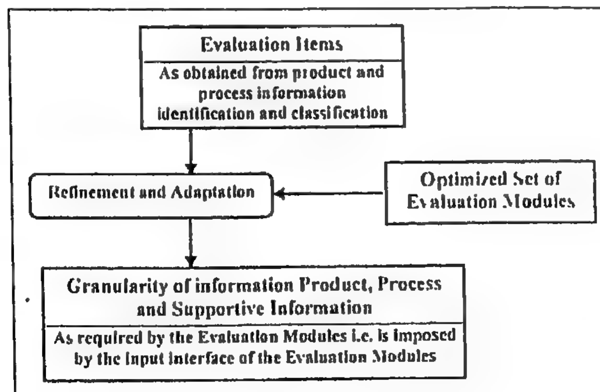


Figure 7: Identification of Granularity of Information .

modules can be substitiuted by a combination. The planning will be done in order to optimize the coverage of evaluation of carrying through the evaluation.

To reduce the set of applicable modules to a minimal set of modules needed for the evaluation, an equivalence relation might be introduced on the set of (all) modules in order to use representative modules which minimize the cost of the evaluation with respect to the applicable evaluation techniques and tools.

In the optimization step, representatives of the applicable modules are selected. Each testing laboratory can use those modules which have lowest cost. With the results of the application of modules, an evaluation statement for the software product can be made.

Figure 6, shows the selection procedure. With information of the evatluation specification, the evaluation items and the evaluation module library, a set of applicable evaluation modules can be identified. In the step "Minimization of Evaluattion Modules" a set of representatives of equivalence classes selected according to the specific constrains of the testing laboratory and to cost-effectiveness.

The optimized set of evaluation modules requires product, process and perhaps supportive information as imposed by the input interface of the evaluation modules. A refinement and adaptation step might be necessary, as shown in figure 7, to relate the information needed by the modules to the evaluation items identified by the information classification.

Planning the evaluation process includes the usual activities to be performed at the beginning of and during a project which are:

- Identifying the evaluation activities.
- Identifying resources (human, tools,... etc.).
- Allocating resources to activities.
- Scheduling and reporting progress.

The evaluation plan including the list of evaluation modules to be applied. Each module includes information from which the cost of its application can be derived. Hence, it is easy to calculate the total cost of the evaluation. However, the testing laboratories act in a competitive market, and therefore the actual price of the evaluation may differ substantially from the calculated cost. Furthermore, in some cases, the cost may also include the whole, or part, of the development cost of a new module.

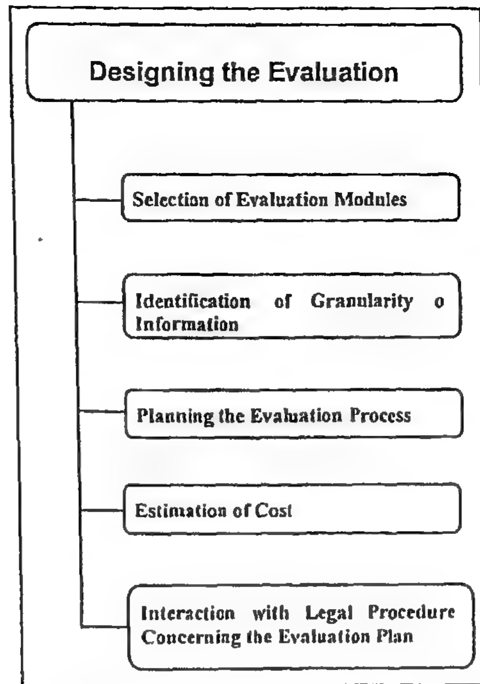


Figure 5: Activities of Designing the Evaluation

After determining the evaluation techniques, that is, identifying which software characteristics should be considered and how thoroughly, and after identifying the technical constraints of the product (like product type, design or program language), a set of applicable evaluation modules are selected from an evaluation module library.

The selection process requires that the evaluation module library be searched for evaluation modules, which will be useful in the evaluation of each attribute for the target product. This directly implies two criteria to be applied in the selection of evaluation modules : **First,** the module must be known and recognized to be useful in the evaluation of the attribute for which it is to be used for. **Secondly,** the module must be applicable to the product part on which it is to be used. For example, many 3GL software metrics cannot be applied to object oriented or rule based software.

However, this set of evaluation modules may not be optimal for carrying out the evaluation. Some modules may be redundant, some other modules may be missing. It must be decided whether new modules must be developed or whether missing

As shown in figure 4, specifying the evaluation consists of the following activities:

- Identification of available documents.
- Classification of items submitted.
- Attaching evaluation techniques to software characterictics.
- Validation of required information and available documents.

The evaluation specifications need not be concerned with whether any attribute can or cannot be measured, i.e. whether particular evaluation modules are available.

Neither should the specification be totally influenced by the wishes of the client expressed in the evaluation requirement. If, at the product analysis stage, some aspect of the product is suspect and is deemed necessary to perform an investigation, then this should be included in the evaluation specification. It will, of course, still be the right of the client to withdraw the product from evaluation if the more extensive evaluation is not agreed.

4.2. Designing the Evaluation :

The aim of designing the evaluation is to produce an evaluation plan by specifying the set of evaluation modules to be used according to the evaluation specification. It describes the selection procedure of evaluation modules, the adaptation of the informtion classified to the information used by the selected modules, the planning of the evaluation, as well as the cost of evalution. The clause concludes with a description of the interaction with legal procedures. Figure 5, shows the activities of designing the evaluation.

*** Evaluation Techniques**

Set of evaluation techiques, encapsulated in evaluation modules, which are to be used in order to evaluate the software product for a software characteristic.

4.1 Specifying the Evaluation :

An evaluation specification is derived by negotiation and agreement with the client of the evaluation and from available information about the software product, i.e. product description, requirement specifications, user manual, etc. Also references to application specific standards and guidelines can be included.

The evaluation specification shall state unambiguously the quality attributes of the software product (functional and non-functional) to be evaluated. The quality attributes of the software product shall be categorised according to the software characteristics given by ISO/IEC-9126. Furthermore, for each quality attribute the required value shall be specified. The list of attributes will be possibly modified by the findings of the product analysis. This list represents the necessary compromise between the characteristics of the product the client feels it is important to evaluate, and a more comprehensive list of characteristics that an evaluator might feel to be the appropriate set of attributes which should be evaluated.

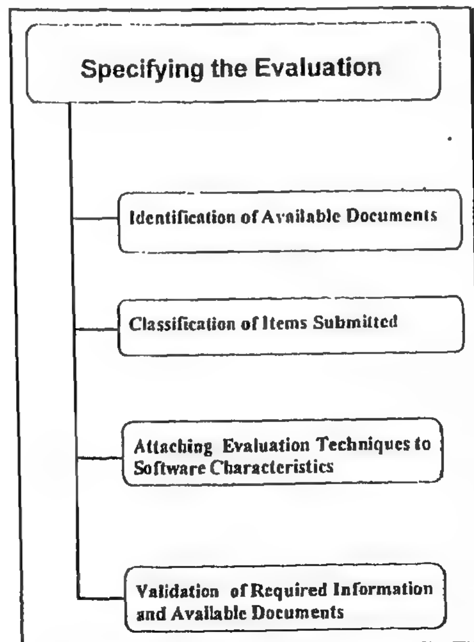


Figure 4 : Activities of specifying the Evaluation.

In the step designing the evaluation process, evaluation modules which define the applicable evaluation techniques are identified and linked to the respective characteristics. The selèction and ordering of evaluation modules and the estimation of costs complete the evaluation design. Performing the evaluation by the testing laboratory, then comprises the application of the set of optimised evaluation modules on the related documents and collecting for each of them the results.

The last step in the evaluation procedure is the reporting of all results achieved during the evaluation requirement analysis, specification, design and performing. The evaluation procedure considers the following basic types of information shown in figure 3:

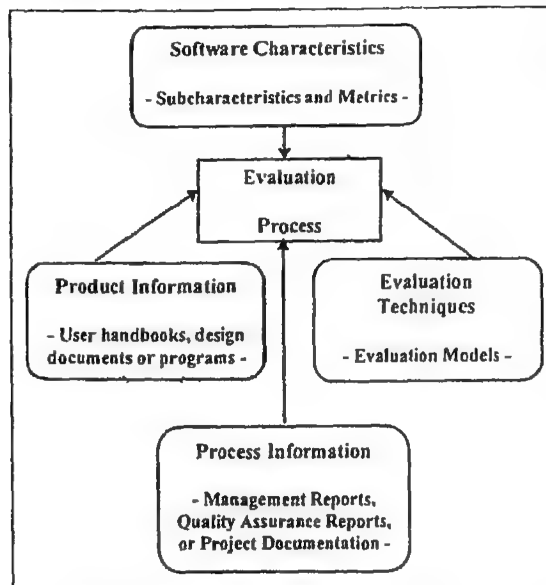


Figure 3 :Information Needed for Software Evaluation

*** Software Characteristics**

Set of software characteristics, which are of interest and should be evaluated. A break down into subcharacteristics and metrics may be helpful for comprehension.

*** product Information**

Including documents such as user handbook, documents, or code.

*** Process Information**

Including docùment such as management report, quality assurance report, or project file.

with their clients, and software users when studying and understanding the results. Evaluation is conducted through a well-defined, step-wise procedure which, to achieve flexibility, is based on a library of evaluation modules. The standardisation of these modules is beyond the scope of this paper.

The procedure is widely applicable for evaluating a software product, i.e. is capable for dealing with any type of software product ranging form off-the-shelf packages developed for a general customer base, through projects commissioned by a customer, to embedded software. This is obtained by using a level approach to quality characteristics and providing a flexible view on classifying product and process information.

The cost-effectiveness of an evaluation procedure is related to the level of confidence required and this level will vary with the critical aspects of the software product usage and the reason for conducting an evaluation. Guidance is provided on how to make an estimate of the costs of evaluation as early as possible in the procedure.

The procedure complies with reqirements stated in ISO/IEC guide 25, which provides general guidance for the operation of testing laboratories and lays down rules for creating a framework for the evaluation of all types of products including software products.

Conformancy testing is not defined in this paper but can be applied if a normative quality model, identifying quality characteristics, metrics and evaluation techniques, is given.

4. OVERVIEW OF THE EVALUATION PROCEDURE

The evaluation procedure presented comprises four steps, which divide the evaluation into distinguished activities as shown in figure 2. Specifying the evaluation finds out the conditions and constraints of the evaluation and defines the objectives in addition to the formal description of the evaluation requirements, consists of the identification of available documents and classification of items received into product, process and (for the evaluation process) supportive information in order to identify which evaluation needs are to be performed. The evaluation techniques are specified in order to determine the information required and identify the items which are missing. They are formal records of the agreement between client and test laboratory of what has to be achieved by the evaluation process. They provide a nominal lists of software quality characteristics to be evaluated at a specific evaluation level and identifies the source of data and evidence to be used in the evaluation process.

Analysability is an attribute of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified. *Changeability* is an attribute of software that bear on the effort needed for modification, fault removal or for environmental change. *Stability* is an attribute of software that bear on the risk of unexpected effect of modification. *Testability* is an attribute of software that bear on the effort needed for validating the modified software.

3. EVALUATION FRAMEWORK ACTIVITIES AND DOCUMENTS

The purpose of a software product evaluation is to provide any interested party with quantitative results concerning a software product that are comprehensible, acceptable and trustworthy. This paper presents a procedure, which defines the activities of analysing the evaluation requirements, specifying, designing and performing the evaluation, and reporting on the results of the evaluation. Figure 2, shows the main activities and documents within the proposed evaluation framework.

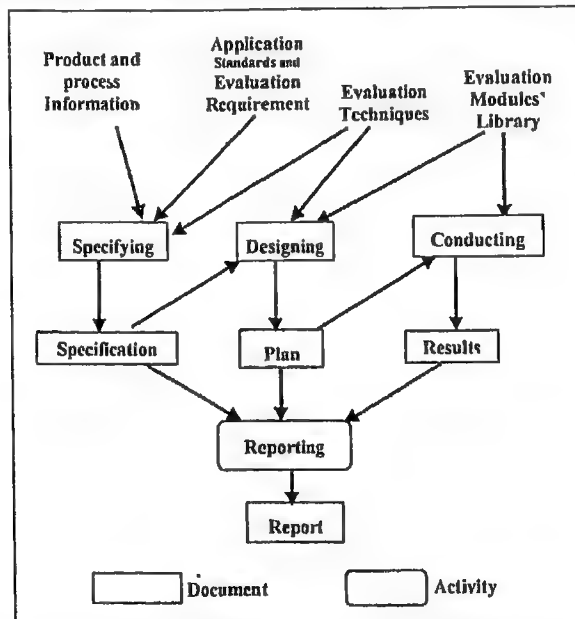


Figure 2 : The Main Activities and Documents within the Proposed Evaluation Framework.

The aim of this evaluation framework is to assist testing laboratories when providing the evaluation services, software producers in planning the evaluation, software customers when specifying certification of the evaluation in their contracts

As shown in figure 1, these sub-characteristics are accuracy, compliance, interoperability, security, and suitability for functionality. Reliability sub-characteristics are fault tolerance, maturity and recoverability. Efficiency sub-characteristics are resource behavior and time behavior. Usability sub-characteristics are learnablility, operability and understandability. Protability sub-caracteristics are adaptability, installability, conformance and replaceability. Maintainability sub-characteristics are analysability, changeability, stability and testability.

Suitability is an attribute of software that bears on the presence and appropriateness of a set of functions for specified tasks. Accuracy is an attribute of software that bear on the provision of right or agreed upon results or effects. Interoperability is an attribute of software that bear on its ability to interact with specified systems. Compliance is an attribute of software that make the software adhere to application related standard or conversions or regulations in laws and similar prescriptions. *Security* is attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.

Maturity is an attribute of software that bear on the frequency of failure by faults in the software. *Fault tolerance* is an attribute of software that bear on its ability to maintain a specified level of performance in cases of software faults or of infringements of specified interface. *Recoverability* is an attribute of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.

Time behavior is an attribute of software that bear on response and processing times and on throughout rates in performing its function. *Resource behavior* is an attribute of software that bear on the amount of resources used and the duration of such use in performing its functions.

Understandability is an attribute of software that bear on the user's efforts for recognizing the logical concept and its applicability. *Learnability* is an attribute of software that bear on the user's effort for learning its application (for example, operation control, input, output).*Operability* is an attribute of software that bear on the user's effort for operation and operation control.

Adaptability is an attribute of software that bear on the opportunity for its adaptation to different specified environment without applying other actions than those provided for this purpose for the software considered. *Installability* is an attribute of software that bear on the effort needed to install the software in a specified environment. *Conformance* is an attribute of software that make the software adhere to standards or conventions relating to portability. *Replacebility* is an attribute of software that bear on the opportunity and effort of using it in the place of specified other software in the environment of that software.

Since developers are responsible for producing software which will satisfy quality requirements, they are interested in the intermediate product quality as will as in the final product quality. In order to evaluate the intermediate product quality at each phase of the development cycle, the developers have to use different metrics for the same characteristic because the same metrics are applicable to all phases of the cycle. A manager is typically more interested in the overall quality rather than in a specific quality characteristic, and for this reason will need to assign weights, reflecting business requirements to the individual characteristics. The manager may also need to balance the quality improvement with management criteria such as schedule delay or cost overrun, because he/she wishes to optimize quality within limited cost, human resources and time-frame.

ISO/IEC-9126 suggest a further decomposition of each characteristic into a set of subcharacteristics. These sub- characteristics are a step closer to the quantitative, technical aspects of the software product. Figure1, shows these sub-characteristics.

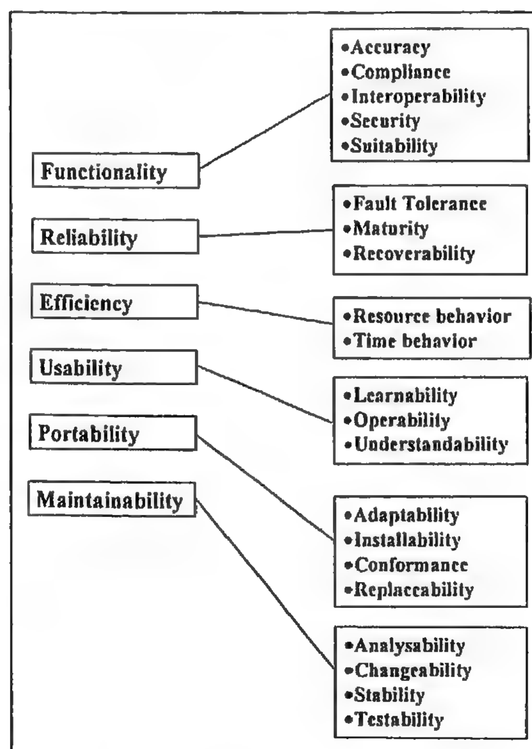


Figure 1 : sub-characteristics of ISO/IEC-9126 standards

2. ISO/IEC-9126 SOFTWARE QUALITY CHARACTERISTICS

ISO/IEC-9126 standard defines six characteristics that describe, with minimal overlap, software quality. These characteristics are **functionality, reliability, usability, efficiency, maintainability and portability.**

Specifying software quality for a product that has still to be developed is difficult for the purchaser or supplier. The purchaser needs to understand clearly and be able to communicate his/her requirements for the product to be developed. The supplier needs to be certain he/she understands the requirement, and is able to assess with confidence whether it is possible to provide the product with the right level of software quality.

Consequently, ISO/IEC-9126 will serve to eliminate any misunderstanding between purchaser and supplier. This improvement in communication will do away with any rework required as result of the software product not meeting the purchaser's requirements. Both the times taken to deliver the specified software product and the cost of development will be lower as a result of adherence to the ISO/IEC-9126 standard.

Functionality is the set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs. **Reliability** is the set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. **Usability** is the set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. **Efficiency** is the set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions **Maintainability** is the set of attributes that bear on the effort needed to make specified modifications. **Portability** is the set of attributes that bear on the ability of software to be transferred from one environment to another.

This decomposition of quality reflects the user's view and introduces the concept of quality in use. Users are mainly interested in using the software product, and evaluate software mostly from the viewpoint of the performance and the service it provides, rather than on the basis of internal aspects or the development process. The process of development requires the user and the developer to use same software quality characteristics since they apply to requirement and acceptance. When developing off-the-shelf software, the implied needs must be reflected in the quality requirement.

software. ISO/IEC-9162 is the software product evaluation standard from ISO. It defines six characteristics that describe with minimal overlap, software quality. It Provides the definition of the characteristics and associated quality evaluation process to be used when specifying the requirements for and evaluating the quality of software products thoughout their life cycle [7].

There is a wealth of knowledge about software quality available [8-17]. The greatest challenge in proposing any model for software product quality is to find a framework that can accommodate this knowledge in a way that is constructive, refinable, and intellectually manageable. The prime requirement of any such model is that it makes clear and direct links between high-level quality attributes and explicit product characteristics at all levels, Beyond this the model must provide:

* Systematic guidance for building quality into software.
* A means to systematically identify/classify software characteristics and quality defects.
* A structure that is understandable at a number of levels, refinable and adaptable.

When carrying out a software products evaluation, clients require reliable results which can be used as a sales aid. An evaluation must be of value to many parties of the software industry, including producers, vendors, users and to the community at large. This means that an evaluation procedure needs to be pertinent, widely applicable and cost-effective. By pertinent is meant that the properties chosen to be evaluated are meaningful to the clients.This means that the purpose of an evaluation, for example in the context of the functionality characteristics, is to assure that the computer program is a trusted implementation of the design specification, which in turn is a trusted transformation of the requirements specification. In other words, the purpose of the evaluation is to answer the question "Is the claimed functionality really available"?. This objective might be difficult to reach, especially taking into account the diverging interests of the client or even the user of the software product.

What kind of guidelines should be provided by the Evaluator's guide? This can be answered by asking the questions "What happens when a problem occurs during an evaluation"? and "What type of problems can occur"?. A missing documents is identified in the evaluation specification. A delayed document leads to a postponement of the respective evaluation activities. If there is a change of product representation, a new evaluation specification has to developed or an existing one has to be modified where applicable (e.g. change of programming language might impose a change of some or all evaluation techniques, which might in turn have an impact on the cost of evaluations). Other problems might be treated in a similar way.

CHAPTER 7

AN APPROACH FOR EVALUATING SOFTWARE QUALITY

Dr. Alaa Mohamed Fahmy

ABSTRACT

In order to takle the increasingly important issue of software quality, ISO/IEC 9162 was developed. This international standard sets out six quality characteristics, which are intended to be exhaustive. In order to successfully and practically apply ISO/IEC 9162, an evaluation framework is proposed in this paper. It is designed to allow technology changes without needing modification of its basic principles. The proposed framework is a four-step procedure for the planning, controlling and reporting on software evaluation. These steps are specifying of the evaluation requirements, evaluation design, conducting of the evaluation and reporting on the results achieved during the evaluation.

1. INTRODUCTION

Significant gains in the quality of software will not take place until there is a comprehensive model of software product quality available. Several different models of software product quality have been proposed [1-6]. While these models offer interesting insights into various apects of software quality they have been strong enough to stimulate siginificant gains in the quality of software or to gain wide acceptance.

Most recently the international standard ISO/IEC-9126 Software Product Evaluation Characteristics (1991) [7] has been put forward as a high-level framework for characterizing software product quality. ISO is the abbreviation for International Organization for Standardization and IEC is the abbreviation for International Electrotechnical Commission. This standard appears to have drawn considerably on the model originally proposed by Boehm [1]. While this standard can provide high-level guidance, it does not go nearly far enough to support building quality into

PART THREE

STANDARDS AND SPECIFICATIONS OF SOFTWARE DEVELOPMENT QUALITY

[14] Picardi, A. "CORTEX Application Factory : concepts and facilities", Auebrach Information Management Reference (Boston, MA: Auebrach Publishers, 1987), p. 3.

[15] Young, J. L. "The software foundry : almost too good to be true", Electronics, (Jan. 21, 1988), pp. 1-6.

[16] Padding, Alan, "Application development: benefits of reuse", www. Informationweek. com (March 31, 1997), pp. 1-6.

[17] Feibus, Andy, "Two paths to RAD" www. Informationweek. com. (March 31, 1997), pp. 53-58.

[18] Florio, Chris, "Authoritative authoring software that makes multimedia happens", New Media, vol. 6, No. 12 (Sept. 9, 1996), pp. 67-75.

[19] "The best HTML tool for you" Internet Magazine, (Feb., 1970), pp. 32-36.

[20] Wiggins, Ricard, "Middleware eases Webmaster's burden" New Media, vol. 6, No. 14 (Oct. 28, 1996).

[21] Spencer, Ken, "Application development : one - stop Web creation". www. Informationweek. com. (April 14, 1997), pp. 1-5.

[22] Wilder, Clinton, Patrizie, Andly and Levin, Rich, "Jave power play", www. Informationweek. com. (March 31, 1997), pp. 14-18.

(3) Functions of large enterprise's products are becoming more complex and sophisticated in an open and changing environment. Therefore, they need applications of diversified nature. Intranet and Web software applications are evolved to meet networking requirements.

(4) Various forms of computer management, control and maintenance systems play important roles in operating and maintaining large software based products. They perform complex functions and usually require a considerable amount of effort to be developed and implemented.

BIBLIOGRAPHY

[1] "Software patents", New Scientist, (Feb. 4, 1993) & Microelectronics Monitor. vol., no. 2 (1995), p. 52.

[2] Humphrey, W. S. and Sweet, W. L. "A method for assessing the software engineering capability of contractors" (Pittsburgh, PA: Software Engineering Institute, Carnegie - Mellon University, 1987), pp. 5-6.

[3] ibid., pp. 23-30.

[4] ibid.

[5] Kellner, M. I, and Hansen, G. A. Software process modelling (Pittsburgh, PA; Software Engineering Institute, Carnegie - Mellon University, 1988).

[6] El Hadi, M. M. "Standardization in information technology and telecommunications for open systems interconnection" In : Towards the development of electronic Arabic resources and the challenges of civilization, (Cairo; Academic Bookshop, 1997).

[7] Caswell, D. L. and Grady, R. B. software metrics : establishing a company - wide program, (Englewood - Cliffs, NJ : Prentice - Hall, 1987).

[8] ibid., p. 25.

[9] ibid., p. 199.

[10] ibid., p. 1.

[11] "Software development". Computer Weekly, (Nov. 7, 1991).

[12] ODA. Information Technology R & D: critical trends and issues, (Washington, DC: Office of Technology Assessment, 1985), p. 76.

[13] Business Software Review, (1987), pp. 29-30.

* reading and writing information to files on server.

(3) Java Language :

In this connection, it should be noted that *Java language* has moved from being an unproved but promising technology to a reliable tool for building business applications on Internet and Intranet [22].

Such Microsystems, Jave soft unit uses Jave for corporate developers. Also, Microsoft is working to encapsulate key portions of Windows 95 and NT applications programming interface as Jave class libraries. This would make Windows programs written in C++, Visual Basic or other Windows development environments more easily portable to Java.

CONCLUSION

As noted above, one cannot ignore the key technology trends that are now unfolding in software production. Large software development firms and organizations, especially in advanced countries, are increasing their degree of software automation and improving their efficiency and performance by better software management practices. Those firms which are existed in Egypt for example, should do some extent acquire, purchase, integrate and exploit software development tools to be able to provide their customers with working software prototypes and customers - oriented software. But this technological capalilities require increasing capital, skilled manpower, access to external knowledge, and an organizational maturity, which are not existed in many software development firms and organizations in Egypt up to now.

The following reasons highlight the importance of technological implications on software industry :

(1) The rapid progress of integrated circuits technlolgy decreases hardware cost steadily. Many hardware components that were developed several years ago may be replaced by off - the - shelf software packages utilizing integrated circuits chips. Some components, which do not have off - the - shelf integrated circuits products are preferred to be implemented in the form of Applications Specific Integrated Circuits (ASIC). These measures also greatly increase the hardware reliability.

(2) The wide applications of various types of Central Processing Unit (CPU) chips simplifies the hardware design in the sense that the same hardware architecture can be used to fulfill different kinds of functions with different specialized implementation of software embedded in the CPU chips.

* Porland's Delphi

* Powersoft's Powerbuilder

These two packages use ODBC, the Open Database Connectivity Standard. Other tools exploit the naive network interfaces of database software itself.

(2) Microsoft's Released Visual InterDev (IDE) : [21]

This software package is an integrated Visual development environment for building and Internet applications. This software package brings together a bundle of visual tools to help Web resources. Visual InterDev shares a common development environment with Microsoft's Visual J++ and Visual C++.

This package was designed to meet the following user requirements :

* An integrated tool for higher productivity,

* Built-in database connectivity tool,

* Built-in development, publishing and site management,

* An open architecture based on HTML/HTTP standards,

* Easy back-end integration with client - server system,

* Open database connectivity.

* Extensibility with other tools,

* Scaleable solutions, and

* A team - based development environment.

One of the important aspects of the Visual InterDev is its active server application development. Microsoft uses the term Web application to indicate any Web site that uses active content on server to perform applications such as pulling information of a database, providing user input to database, and dynamically generating HTML in response to user.

Web applications that are built with Visual InterDev typically use the Active Server Pages (ASP) component on server, which contains server - side scripts (either Vbscrits or Jscripts) along with HTML. The ASP pages can perform almost any task that the traditional application can do, e.g. :

* Integrating with database;

* displaying information to users;

* accessing external server components; and

* **Web Top Publishing (WTP) :** This category currently has only one package, NetObject Fusion.

In addition to HTML Tools which are very crucial to network applications, there are some developed software packages which are of interest to Internet Web and Intranet applications, such as :

(1) Web middleware :

This technology has evolved to provide help to those who may have a database that needs a Web front - end [20]. These software packages sit between Web server and SQL database, eliminating some of the drudgery of Common Gateway Interface (CGI) programming while improving productivity in applications development.

CGI developed from a need to extend Web beyond simple delivery of HTML documents because enterprises want to be able to process information submitted by Web users in some intelligent way, Also, middeware stems a need for a mechanism to connect a person running a Web browser to programs and files on the server end.

CGI was evolved primarily under UNIX, but was later translated to Windows and Macintosh server environments. This defines how a browser user's input communicate from a Web server process to an external program or script. Using CGI, Webmasters could enrich their sites, connecting users to live data, whether it was from a product database or a weather map.

The problem was that Webmasters had to write separate CGI programs for each task in scripting languages such as C, Visual Basic, and Apple Script. These programs or scripts accomplished similar functions but rarely reuse code. Webmasters needed to live databases, can provide important performance advantages over CGI. The traditional CGI programming uses a new process to be spawned for each invocation of a given script.

For server handling few transactions, this is considered fine, but those handling hundreds of transactions per hour can slow to crawl as memory fills with redundant process. Most Web middleware software settle this problem by lauding a single program, a server plug-in, that remains inconstant communication with the server no matter how many transactions are outstanding. Commonly supported servers include Netscape, through its NSAPL, and Microsoft's Internet Information Server, through ISPAPI.

Therefore, Web middleware products are tools which are considered as rapid application development solutions. Examples of these tools are :

* Has direct support for movies and multimedia files, such as Shakewave, Quick Time and AVI. It is also required to look for image previewing and management features.

* Other useful features to look for are spell checking, syntax coloring (the color coding of HTML tags and attributes), and the ability to edit multiple documents.

* Ideally a software product should have also HTP (and helpfully HTTP) transfer support.

* It should have an uninstall option just in case the developer changes his mind.

It should be noticed that HTML tools range from single text editors to DTD - like programs which give complete graphical control over page design. Deciding which package suits the needed page, the developer will depend on his skills as a programmer and a designer in the meantime. For example, designers will appreciate programs such as NetObjects Fusion because it gives them a DTP - level control over appearance of Web site. On the other hand, programmers will probably prefer packages such as HotMetal Pro which gives them access to the latest HTML and Web site gizmos like ActiveX controls and Java applets.

If someone is working with Web design team, he probably needs more than pick of software. He will need different tools for different members of the team carrying out different tasks.

The current version of HTML is 3-2. The latest update to the standard have essentially been pushed through by Microsoft and Netscape as they constantly revise these browsers.

The version 3-2 of HTML Document Type Defintion (DTD) includes most of these Microsoft and Netscape extension alongside with minor revisions to the structure of some of the new elements to improve cross - platform and cross - browser compatibility. Both Navigator 3.0 and Interment Explorer 3.0 can display documents created with the latest version of HTML.

The information system Web publishing tools or HTML Tools come in four different types which are :

* **HTML editors** : the developers works, here, in text mode and have buttons to add to HTML tags.

* **WYSIWYG editors** : These products work much like word processors, so it is not needed to edit HTML at all.

* **Site Management Tools :** These products focus on the creation and management of Web sites rather than single pages.

Therefore, the right authoring environment for these tools is crucial, learning curve, playback and authoring platforms, and project are some of the factors to consider.

There are several mid - to - high end authoring programs, mostly timeline - and icon - based for Windows and Macintosh platforms. All the existed programs combine graphics, sound, animation, video and text into projects with more sophisticated synchroniztion and interactively than is possible with multimedia presentation programs such as Astound and Power - Point. Some such as Macromedia's Director and Advanced Media's Media Master Pro, let the developers to create and edit media right in the applications. Others, such as Apple Media Tool, work only with media created elsewhere. All except Scala Multimedia MM100 let distribute the work with run time player at no charge.

While Director is still the dominant player in the authoring market, most of the applications which were introduced in 1995 are intended to chip away to various niches.

For example, Innovus Multimedia program is geared toward business applications, while less emphasis on fancy graphics and animation and more focus on ease of use database connectivity and network distribution. Media Master Pro, on the other hand, bundles various media editors for an integrating authoring solutions, and Scala Multmedia 100's distinctive multitasking operating system provides the smoothest playback of Windows applications. Finally, looking Glass Software's Media Verse is a good choice for text - oriented applications, and Apple's Media Tool takes ease - of - use honors. and is also one of the best choice for cross - platform work.

5. INTERNET APPLICATIONS SOFTWARE :

The following presentation reviews in a brief manner some aspects of software packages related to ***internet*** and ***intranet*** or web applications. The HTML tools and Web middleware, Microsoft's Visual InterDev and language are introduced.

If it is needed to build a Web site, the right software should be allocated to help in the creation of the required site. But just which of the more than 140 software products [19] be selected and used to design the site?

There is a core set of features that it is important to look for in a Web publishing packages. The chosen software package should do all the following :

* Supports the HTML 3-2 standard, and has the flexibility to handle future changes to HTML standards.
* Handles Java applets, ActiveX controls and scripting and image - map creation. It is needed also to look out for the ability to graphically edit forms, tables and frames.

(2) ***Delphi 3 Client / server Beta*** : released on May 1997, Porland is doing its own matching, adding ActiveX and distributed applications development capabilities. The main features of this package are :

*** Descrption :**

Porland International's multifier development tool based on object - oriented Pascal.

*** Strengths :**

- Easy to use interface,
- More - mature distributed object technology,
- Object paradigm and repository encourage code form reuse.

*** Weaknesses :**

- Object Pascal has a sleeper learning curve than the Visual Basic.
- Manual migration path for converting applications to ActiveX documents.

The two releases raise the bar for Powersoft's Power Builder 6.0. They also finally offer Windows developers some solid options for rapidly building and deploying distributed applications across their networks and intranets. The improvements in these tools may even be enough to lure developers away from Windows C++ tools.

Also, both tools make it easy to create ActiveX controls and servers, ActiveX documents that can be run from within a Web browser, and executable applications. Both tools can create applications with distrbuted components using Microsoft Distributed Component Object Model (DCOM). Also, both tools support only development of 32-bit Windows 95 and Windows NT applications. 16-bit users of Windows 3.x are left behind to use only older versions of these tools.

In fact, both development tools are very similar when it comes to developing stand - alone a simple client - server applications. The real differences between these tools emerge when they are applied to the task of enterprise wide application development, then the gap between Microsoft's and Porland's strategies for distributed development becomes visible.

4. AUTHORING MULTIMEDIA SOFTWARE :

Multimedia authoring spans a huge range of projects from presentations, computer - based training programs and product demonstrations, to entertainment, and education CD-ROMs, Simulations, Interactive Web sites and Space - alien - destruction games [18].

The best practice for software reuse involves the following characteristics :

* Build a strong architecture that provides the framework for reuse activities,
* Use a software - development process that promotes and controls reuse,
* Reuse more than just code, such as product and project templates, models, requirements, specifications,.. etc.,
* Practice domain development, such as identifying subdomains for a line of products, each with significant reuse potential. This helps reduce the scope of software development to a manageable size,
* Integrate reuse into project management, quality initatives, and software development. activities,
* Emphasize partnering to achieve reuse across product boundaries,
* Use automation to support reuse, including automated process improvement, software repositories, configuration management, domain development engineering, software factories, and integrated CASE.

3. RAPID APPLICATIONS DEVELOPMENT TOOLS :

In spite of there are many rapid applications development tools, only two of them which are widely used are being discussed in this paper. Despite their parallels, Microsoft's Visual Basic and Porland International's Delphi as rapid applications development tools, they are clearly oriented toward different audience [17]. Visual Basic is aimed to developers who are using Microsoft products almost exclusively. Delphi provides similar (and in several cases better) functionality for distributed applications, In the same time the two have never looked more alike.

(1) *Visual Basic 5.0,* released in March 1997, brought the Visual Basic line even closer to parity with Delphi's performance by way of its long - awaited code compiler. The main characteristics of the Enterprise Edition, are as follows :

* **Description :**

Microsoft's development tool for a distrbuted database applications using Microsoft's products.

* **Strengths :**

- Easy to use, with gentle learning curve.
- Native code compiler debugger for rapid code fixing.
- Includes also, interpretive debugger for rapid code fixing.

* **Weaknesses :**

- Immature distrbuted object implementation for enterprise class applications.
- Database development tools do not do much with other vendors' databases.

ware execution speed. Prototype is built, and software developers can see how code modification will work on the proposed hardware. Such tools are operating on mini- and mainframe computers [15].

Automated tools alone are not significant to ensure productive software environment. With no coherent methodology standards and set of controls, automated tools, sometimes, are just merely help developing a software project. Hence, they can also create an image of modern methods and high technology without substantively increasing productivity.

Successful use of automated tools requires a fit between the tool and the environment in which it will be used. The selection of the tools can be a complicated and confusing process. Tools must, in some ways, be similar to what a development company already does and knows; and they must be supported and maintained.

2. *SOFTWARE REUSE :*

If the programmer gets the software right for managing the movement of some thing from one place to another, it can be reused effectively regardless of what is being moved and how it is moving.

while reuse comes primarily through planning, design and organizational efforts, new technologies such as software repositories and components can encourage software reuse. Components which are defined as reuse software services bundled into an independent package, have been hailed recently.

Texas Instruments Inc., in Dallas, Texas is considered, for example, a leader in component reuse, identifies five types of components [16]:

(1) User interface controls, which are the technical widget, such as OLE controls,

(2) Technical infrastructure components, which include some bundled functionality, such as Internet connectivity,

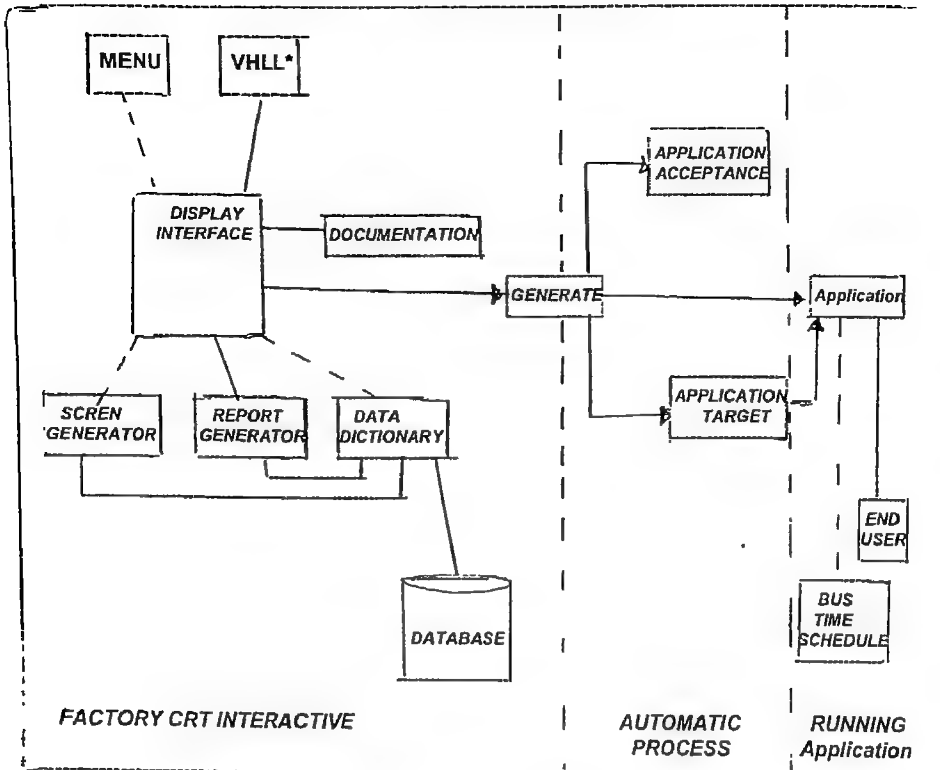
(3) Domain - generic components, such as the corning authertification service which provides functionality across the establishment,

(4) Domain - specific components, which provide a complete package of business functionality for a partiular industry, such as telecommunications billing,

(5) Industry applications consisting of complete systems or templates for such functions as materials management or general ledgers,

It is relatively cheap and easy to get reusability for the first two types of components. Almost any developer can figure out how to use a package OLE custom control to create a sophisticated data window or provide some advanced functionality,such an image editor. The components do not require a lot of up - front planning, and they are readily available at low cost nowadays.

Fig. (4) : The Cortex Application Factory



* VHLL = Very High Level Language

Anyway, computer - aided software engineering, i. e. the CASE tools are available nowadays, for each of the life cycle development stages such as requirements analysis, detail design, coding and testing. However, these tools are not able, up to now, to replace or take over all the manual work in each of these stages. A reasonable approach is to combine these tools with the manual work, so as to maximize efficiency and improve the design quality.

(3) Integrated set of Tools :

Some tools now link hardware and software design (e. g., an integrated set of tools such as : C compiler, assembler, linker, simulator and debugger). These tools are developed by many companies such as Microsoft and Quantitative Technology Corp. (QTC), For example, the Q TC allows its software developers to evaluate design interactions of both hardware and software without waiting for a final version of either. A simulation informs hardware team how design changes will affect soft-

102

(2) Screen and report painters for prototyping user interfaces.

(3) Integrated dictionary for storing and cross - referencing all systems analysis and design information.

(4) Automatic checking and reporting the completeness and consistency of structured diagrams.

(2) The Cortext Application Factory :

This tool is used by systems analysts and programmers to develop and maintain medium - to - large information systems. It is, also, used to lead a developer through the steps of software development process and includes features similar to those of "Exceleratoes", including :

*** screen and report painters for prototyping user interfaces.**

*** A dictionary for storing documentation about a system.**

*** Automatic checking for completeness and consistency of program specifications.**

*** A code generation capable of automatically generating about 95% of the program code from program specification.**

*** Automatic program documentation generator.**

Once accepted, the applications are translated into machine code by an optimizing compiler for fast computer run time and greater machine efficiency. Applications developed with the Cortex Applications Factory range form such applications as sales tracking, order entry, accounting, inventory and payroll to more sophisticated ones, such as regional price trends, tracking for commodities and orders for materials, etc. Also, applications range in size from a few files, screens and less than 50 database fields to those with several hundred files, screens and thousands of fields [14].

the developed applications generators can produce major benefits in productivity and case of implementation, use and maintenance if accompanied with good training and providing unexpected problems do not occur. Quantitative benefits include reduction in system life cycle costs and rapid development prototypes to ensure quick turn around for end user assessment. Among the possible quantitative benefits are the generation of documentation directly for specifications rather than program code, rapid iterative prototyping and greater control over maintenance.

Automated software documentation management techniques are gradually being introduced to software developers to improve quality of both software and documentation.

to develop software one instruction at a time. Applications' generators are also valuable for their ability to develop quick prototypes of a desired capability.

The emergence of application' generators oriented around a database management system (DBMS) and report - generation capability has created an attractive approach for both software productivity gains and for software customization. Some software development firms in advanced countries which are using these tools have improved their productivity by factors of two as much as 20 across various phases of the software life cycle [13].

The facilities in three different tools : one for complex, real - time computer systems, the others for business applications are presented here.

These tools are reviewed briefly to illustrate the variety of ways in which software automation currently increases the productivity of software of all sizes and types.

(1) Auto - G Tool :

This development tool is produced by a small U.K. company, Advanced Systems Architecture (ASA). It is so far the only non U.S. system design tool that has been evaluated and supported by U.S. Strategic Defense Initiative (SDI). The ASA company has focused its operations on high reliable, high security real - time computer systems.

Auto - G tool uses a formal graphical notation, G, to enable software developers to build a system on a workstation in stages, from requirement specifications to code generation. When the design is complete, the Auto- G toolset provides a code generation that converts detailed design automaticatlly into a variety of programming languages, including for example, C and Ada languages. The Auto-G provides and includes the folowing :

* **Menu-driven, on-screen selection of design symbols,**
* **On-screen editing,**
* **Full graphical manipulation,**
* **Hard copy output low - cost graphic plotters, and**
* **Automatic checking of logical consistency of design.**

The excelerator of this tool has four basic facilities for autmating system analysis and design tasks :

(1) Automatic diagramming tool for drawing structured diagrams, such as Data Flow Diagrams (DFDs), structured charts, data models and control flow diagrams.

Metrics will not take hold in a firm or company unless they are supported at the top of the organization. Senior management must understand the benefits and campaign on their behave, supporting middle management who can otherwise find themselves in an awkward position. But as this top level support, metrics need a sponsor group. It is recommended that an entity whether governmental or non-governmental should allocate full - time to tasks of promoting metrics and overseeing their collection, a status which is not existed also in Egypt up to now. We recognize that the importance of the existence of quality standards which provide the guidelines and criteria for software producers and can be quantified using metrics techniques.

THE IMPACT OF NEW TECHNOLOGIES ON SOFTWARE DEVELOPMENT

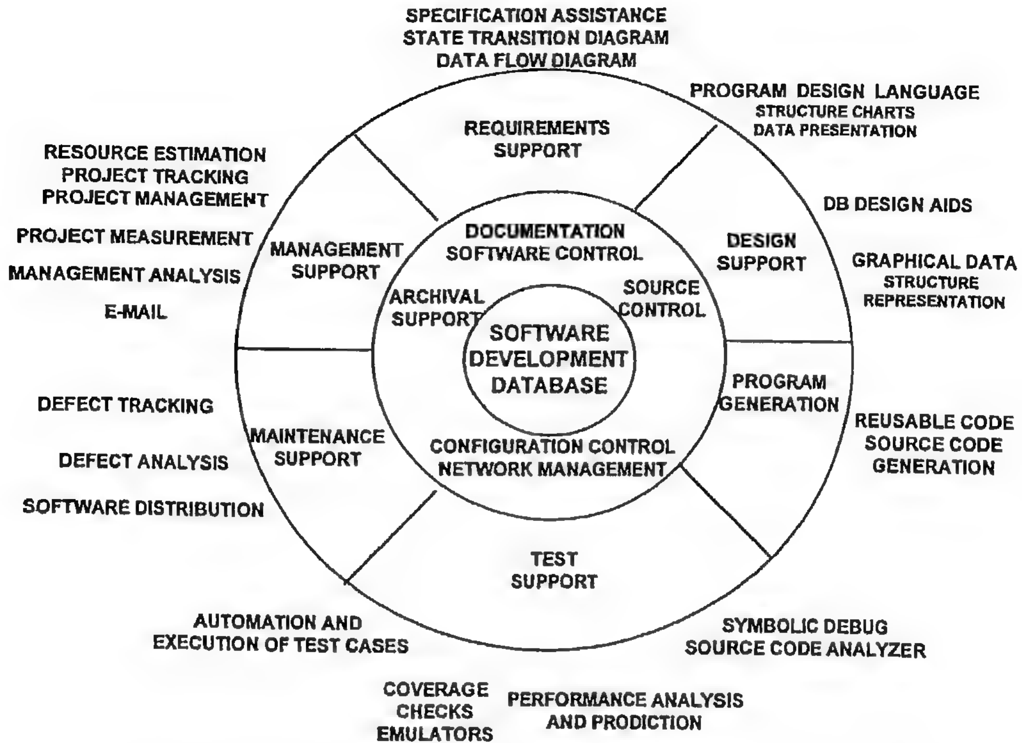
Research and development efforts in software development engineering have produced new methods that show promise for improving programmers productivity and software reliability. As the software development industry gradually becomes less labor - intensive over time, the quality and availability of skilled labor will become more important than labor costs. Pressures on the quality of software labor markets will increase competition among software firms and organizations, for a skilled labor in software production. But the introduction and adoption of these new technological tools and techniques is completely difficult and expensive, and they are thus adopted bit - by - bit as project budget allows [12].

1. AUTOMATED SOFTWARE DEVELOPMENT AIDS :

The conventional "Waterfall" software life cycle has been modified since its inception in the early 1970s. The major stages of software production (specification, design, coding, testing and maintenance) remain as planned, but efforts are underway to alter and automate many aspects of the software life cycle. Lower development and life cycle costs are being realized through better structured and documented software, program support library procedures, diagnostic aids, environmental simulators, test data management systems and applications generators. Such tools and techniques support different phases of life cycle. Some tools support early phases in the form of automated diagram drawing, screen painting, and error checking. Others give assistance by automating code generation and documentation.

Sizable applications programs can be generated using a very small number of user - language directives, which means that developing software with applications' generators can be a far more cost - effective pursuit than hiring programmers

Fig. (3) : Hewlett - Packard's software engineering productivity environment



Successful integration, collection and use of software metrics into software development process is the main objective of software firms and organizations using such tools. The time factor involved in learning tools and incorporating them into the entire development process has been the primary obstacle of time implementation. For example, the history of HP's Software Metrics Council shows that it took roughly three years of collecting and analyzing data using software metrics before there were sufficient data available to show measurable trends for the entire firm [10].

Software productivity is improving at a modest rate of about 4% a year compared with hardware, which is doubling in performance every four years [11]. To identify and promote best practice in software metrics, a project called ESPRIT brought together nine companies who have pioneered the use of metrics in Europe. The group of companies had significantly collective experience to offer. The results of this project were included in an interim report which describes an approach to metrics using current best practices.

Documentation is frequently a low priority activity in software development process. Yet in the studies at IBM, the overall quality of software is very much a function of documentation. The above figure suggests that 25% of all software defects were in documentation delivered to customers [8].

Many tools are available for software metrics, some are in public domains to assist in collecting and examining measurable results during the various development stages; among of the already extensive utilized ones is the program called SOFTCOST which is used by the U.S. NASA's Computer Software Management and Information Center (COSMIC) which had developed and offered more than 1100 computer programs. The SOFTCOST package estimates software size, implementation productivity, recommended staff level, proper location, amount of computer resources required and cost of software documentation. Also, this program was designed to provide project managers with a comparison between their expectations of a project's development and industry - based statistical expectations of the project. The software metrics data provides an additional basis for budgeting time and effort to a project.

A complete integrated set of metrics' tools does not exist to be utilized by large software firms in their development processes in Egypt, up to now.

Anyway, the following figure (3) illustrates what Hewlett- Packard Co. considers an ideal software development environment which is reinforced by tools and metrics. The central database provides a common control mechanism for all important parts of a project [9].

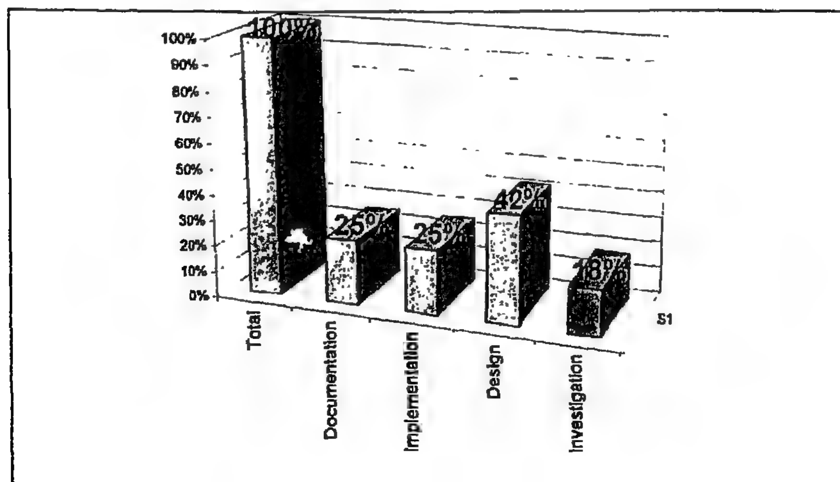
2 . ASSESSMENT OF SOFTWARE METRICS :

The status of software metrics, or measurement, is still somewhat immature and imprecisely defined, in the case of Egypt for example [6]. Nonetheless, measuring software products and software development can lead to substantial benefits for reasonable costs. Short - term productivity improvements, and the establishment of a common development environment, have been reported by companies that use software metrics. To many project managers in software firms, software metrics are a means for more accurate estimation of project milestones, and a useful mechanism for monitoring progress [7].

Basically, ***software metrics*** are a way of measuring the various attributes of the software development process. Attributes include the size of a program, its costs, the number of programming errors or defects, the level of difficulty of the project and the methods of communications required between members of the porject team.

The following figure (2) shows an example of using software metrics in software development in what is usually noticed as an ignored activity is documentation.

Figure 2 : Introduction of software defects



ASSESSMENT OF SOFTWARE STANDARDS AND METRICS

1 . ASSESSMENT OF SOFTWARE STANDARDS :

Standards for software engineering will take a long time to develop. But with the rapid technological improvements and increasing competitiveness characteristics of the software industry, government agencies and other large purchasers of software are using new techniques to evaluate contractors' abilities to develop software packages according to modem software engineering tools and methods. Also, a great number of software firms and organizations are starting using new evaluation techniques to asses their own ability to create critical software projects and assess their overall competitiveness.

One methodology, developed by the Software Engineering Institute of Carnegie - University for the U.S. Air Force [4], examines a contractor's capabilities in :

* ***Organizational and resource management,*** including software engineering training and the adequacy of support facilities. This also includes, quality assurance, process management, configuration control, and quality and quantity of resources.

* ***Software engineering support and management***, which includes the scope and use of conventions, formats, procedures and documentation during the software development phases, software quality assurance and data management. It also, includes the depth and completeness of the process and how it is measured, managed and improved.

* ***Technologies and tools,*** a contractor may use in the software engineering process some computer software development tools to cross - reference between modules and to design debug code.

Since this methodology has become only one additional part of criteria for procurement, it is unlikely that it will create a radical change in contractors who are selected for software development projects. But the evaluations will probably lead major software firms to manage their software development process more closely.

An awareness of software producers of these evaluation methodologies is important to future development.

Raising some of the questions presented in various methodologies, [5] may improve the software process in software development firms and organizations, and to some degree enhance their competitiveness in international export markets. These methodologies are usually concerned with standards and practices of the software firms in the preceding areas of contractor's capabilities.

* Leverage existing processes and minimizing some areas of change.

* Make the contributions of everyone in the development team very visible.

Therefore, managing change requires good communication of change, acknowledgement of the progress to be made, and encouragement toward the optimum goal. After each software life cycle checkpoint, the quality assurance team performs an audit by interviewing the team members using checklists as mentioned in the requirements phase. The software quality assurance members are actually considered the process consultants. They bring experience and maturity to the audit interviewing, reporting and follow - up consulting.

These audits have several major benefits :

First, The project team should know ahead of time that a process they are using during a phase would be objectively evaluated by independent team or firm. This has the effect of evaluating the importance of using a defined process;

Second, the audit interviews could cover critical issues and risks for the software development's products that are almost always a result of deviating from the project's plan or processes.

For example, during the audit a problem with inadequate staffing for test planning activities could be identified. The project plan calls for completion of test plans for the context feature changes before the design complete checkpoint. Therefore, the issues and risks should be reviewed by the management team and decisions are to be made to take corrective actions.

It is believed that the circumstances and development environment for the firms and organizations are not unique. Many firms are trying various quality improvement endeavors for software development and finding it very difficult to make the changes necessary to be more rigorous and disciplined in their development practices. It is also, believed that many of the essential ingredients to make a software improvement program should reduce the time span of the development cycle with utmost efficiency and reliability.

Therefore, the capability maturity model (CMM) provides an excellent framework for defining software development process improvement for small software development firms. Achieving level - 2 of the CMM status has largely been a matter of establishing a direction with leadership from top management than instituting a credible program for improving the practices used by software projects in planning and managing the work according to the guidelines of tthe CMM.

** External commitments are reviewed and approved by the business team managers.

** Responsibilities are assigned for developing and maintaining the design phase.

** Adequate training and tools are provided for planning the design phase.

** A design phase plan is prepared according to document procedures.

** Design phase activities are documents in the design phase plan.

** Estimates of size, effort and cost for design phase planning are developed.

** Requirements are traceable in the design plan.

** Issues related to design are reported and tracked.

** Design phase plans are to be updated to reflect changes in requirements.

IV. DESIGN PHASE :

During the design phase of the software life cycle, the key process areas are considered the practices in project tracking (i. e., managing and controlling work products, especially changes in requirements), and in project planning for this phase. The success in applying the software life cycle and the level - 2 of the CMM model processes to this phase of work is evident from the following two major accomplishments: first, the team should complete every aspect of the design phase including reviews and inspections. In the past, design specifications and design reviews were cut short because of schedule pressures; Second, the team should be able to make decisions early in the project about eliminating features that would be too costly to implement.

For example, unit testing capability is a feature to be considered for the software development, but it is eliminated during the requirements and design phases because of the staffing trade- offs. This action substantially reduces the risk of schedule slippage later on.

V. IMPLEMENTATION PHASE : MANAGING THE CHANGE :

Getting the software development firm to adopt the changes necessary for the CMM compliance is considered an important step in implementing the new processes. Defining the new policies and procedures and team providing training is necessary, but not sufficient. Changing the processes involves changing the culture of the firm, and this is where it is critical to get everyone thinking about the changes in a positive way. This aspect of change management is approached by deciding what would be tried to accomplish the following goals :

* Demonstrate success with a phased approach.

III. REQUIREMENTS PHASE :

During the requirements phase of the software development life cycle, the key process areas to work on are practices for requirements management and project planning. By using readily available customer survey data, structured processes, management review milestones, and training, it would be able to reduce the time allocated to perform this phase. It is important to gain a deeper understanding quickly of the new features demanded by the customers, and to translate this information into work steps that would be needed in the design phase afterwards.

One of the problems that the requirements phase is to face reducing the list of possible things to be accomplished to a few high impact requirements that are to be accomplished within the schedule. Thus, to collect requirements information, survey questionnaires based on a user - centered design methodology are to be created to facilitate rapid feedback from customers using telephone surveys for example. The process consultants designed standard templates for developers to describe the customer requirements.

One criterion for determining what new features to include is an estimation of resources and effort needed to design the new features. The decision making, process is very essential element for narrowing the scope of the project and finalizing the work commitments for the next phase of the software development.

A ***Software requirements checklist*** should be planned. A portions of this audit checklist for requirements phase which is used to assess the software development firm or organization life cycle against the level - 2 of the CMM model contains the following factors :

** Responsibilities are to be assigned for planning requirements phase activities.

** Staffing is sufficient for the planned activities.

** Adequate training and tools are provided for planning activities.

** Requirements phase activities are documented in a requirements phase plan.

** Estimates of schedule are prepared and included in the requirements phase schedule.

** A software life cycle is identified or documented in the requirements plan and design phase.

** Work products for control of the project during the requirements phase are to be identified in the project planning documents.

** Commitments are to be negotiated with the business team managers and other affected groups.

ware development teams. Communicating what is expected, describing changes from the way it is used to do things before, and providing group of individual training in new methods are considered very important steps in deploying the process changes. It is known, that the development project teams need to understand the rules early, so that the project could proceed smoothly.

During the ***assessment and planning*** of the improvement development process project, it is recognized that many practices could be used at level - 2 of the CMM compliance. Therefore, it is important to leverage as many of the current practices and procedures to minimize the effort required and reduce the amount of change being introduced. For example, standard software life cycle development toolset, good practices and tools in configuration management, defect management, inspections, coding and testing are among the practices to be used. Also, a customer satisfaction survey and software metrics are of much help.

The development environment consisted of a suite of tools integrated together with the main development software tool such as CASE, ITE or SoftBench of HP are much needed as a rapid prototyping tool. this tool is to provide the framework of editing, compiling and debugging. Capabilities of these tools are also being used to assist in program understaning through all graphs and documentation of object oriented designs of the new features. Integrated, in such development tools, is a configuration management tool for managing source code, and the ability to access defect tracking. Having these tools is considered a major factor in maintaining developer productivity while making process changes in other areas.

Inspections and **software** metrics should be established as a part of the organizational culture. Although, both of these areas are considered elements of the level - 3 of the CMM, they are also, required for level - 2 to maintain the benefits which are achieved project with these practices. Because inspections and metrics (defect tracking, schedule slippage, and effort reporting) are institutionalized in the engineering and software management practices, it is recognized that this would be distinct advantage for the development of the software maturity process of level - 3.

Also, in the planning phase, it should minimize changes within the areas of software configuration management, i. e., software quality assurance, subcontract management.

Therefore, methods for measuring the progress of the software project development are very important to be established in this planning phase.

The only way to do this is through auditing the phased stages of the software development after each major checkpoint.

* May specify who does what when.

Checklist * Specifies what or how in abbreviated format,

* A short form of procedures for easy reference or verification of actions.

Template * Specifies the content or quality of what will happen,

* Provides guidelines for creating necessary work products.

Training * Provides organized information on processes that individuals need to perform their jobs,

* Covers policies, procedures, checklists, tepmplates, or instructions,

* May be formal classroom or informal orientation.

Work Product * Specifies a plan or results,

* Created as an output of applying a defined process,

* May need to be "managed and controlled".

II. FORMAL PROJECT PLANNING AND DECISIONS PHASE :

To give this improvement project the greatest chance of success and reduce the overall risk for the project, a formal project plan is to be developed covering every phase of the defined work and the timing for deployment of the processes. This plan is to review and to be approved by the team's members before beginning the implementation.

Several key decisions need to be made about how the project deliverables would be designed, reviewed, approved and deployed into the software development operations. Also, several models for the process creation, approval and deployment are to be examined and discussed before it is decided to use a definedeploy approach to be mapped into each development life cycle phase (i.e., requirements, design, implementation and test). This could provide a structure within which process deliverables could be refined and improved.

The software development process needs to show results as early as possible to capture the attention of the organization, and to keep things focused on process improvement. To accomplish the objectives of the early showing results, the deployment stages of the development process are to be designed. This tactic provides visible results and feedback to the organization by coinciding with the life cycle phases.

A series of short steps for defining, reviewing and approving the policies, procedures and templates to be followed are needed before deploying them to the soft-

* Are the project's planning activities and deliverables tracked, (e. g., schedule, effort and tests)? Fully or Partial.

* Are the actual results compared to the estimates throughout the project? Fully or Partial.

* If there is a variance, does someone take corrective action or explain why variance has occurred? Fully or Partial.

* Are changes to the activities, deliverables and schedule discussed with the people who will be affected? Fully or Partial.

* Does someone review the project results regularly with development team? Fully or Partial : or Almost Always, or Often or Seldom.

Therefore, in applying the capability maturity model effectively, one of the critical steps in getting started is to understand the model in detail. The specifications of the model are to be reviewed, the interpretation should be developed. and the model itself should be translated into a language to be applied by the firm. This is considered an important step because the development maturity model contains many practices that are better suited to large firms, and it is necessary to interpret which of these practices would apply to the firm. Also ideas of how best to deploy these practices should be developed.

Several process assets of the development firm could be leveraged to support the key process areas requirements. These assets are software live cycle; formal documentation for patches, releases and defect management procedures; and several informal methods of documentation for specifications, design and testing.

A need for a ***process architecture*** is to be needed when it is attempted to describe what deliverables would be needed to support the definition of processes. A process architecture is analogous to other types of professions such as engineering architecture (e. g., buildings, software, hardware, etc.).

Therefore, a process architecture describes the layout of computers needed to build a system for a specific purpose.

The elements of a process architecture are as follows :

Policy * Specifies what will happen,

* Sets the cultural expectations,

* "That's how the firm does things".

Procedure * Specifies how it will happen,

* A set of steps for doing something.

tribute to satisfying the intent of these attributes for each key process area. Any firm or organization can use alternative procatices to accomplish the goals of the capability maturity model.

III. THE CHALLENGES OF THE CMM :

It usually takes most software development firms about a considerable time (about 2 - 3 years) to go from level -1 to level - 2 of the capability maturity model's compliance. However, based upon sound business reasons, the general management of the firm could commit itself to reach level - 3 of this development model in less time (about a year and half only) and assign few people to do the needed tasks.

During the planning stage, it could be discovered that because this is a reasonable development process program, level - 2 could be completed in less time with less people than the previous practice, because the CMM provides immediate benefits to the firm which applies it.

Therefore, the software product development team could reach level - 2 of the CMM compliance within a few time of staring the project, beginning with investigating the project and continuing implementing it. Throughout this period, two deployment cycles with the development team could be completed : the internal audits of the team processes verify what is being operated at level - 2 of the CMM, and several more audits of all the development team are executed externally.

SOFTWARE DEVELOPMENT LIFE CYCLE

I. INVESTIGATION PHASE :

To investigating the current practice, a technique called **Software Process Profile** could be used. This technique which was developed by Hewlett-Packard and the Software Engineering Institute at Carnegie - Mellon University, is considered an assessment of the state of an organization's software development process which identifies strengths and weaknesses, highlights the process improvements which the organization values most, and recommends areas for change. The profile uses the capability maturity model as a standard for evaluating the software process. Through the use of questionnaires and open-ended interviews, the profile could provide results for the key process areas of the software development maturity model.

To complete the profile, developers and managers of the project and the firm must fill in a questionnaire that is to evaluate the firm's software process maturity against the capability maturity model's requirements. The results of this questionnaire are to be compiled into a software process profile for the firm. An example of typical survey questions to assess the software project tracking and oversights are :

*** Requirement Management :**

1 - Requirements are controlled to establish a baseline for software engineering and management.

2 - Plans, products, and activities are kept consistent with the requirements.

*** Software Project Planning :**

1 - Estimates are documented for use in planning and tracking.

2 - Project activities and commitments are planned and documented.

3 - Affected groups and individuals agree to their commitments related to the project.

*** Software Project Tracking :**

1 - Actual results and performance are tracked against the software plans.

2 - Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the plans.

3 - Changes to software commitments are agreed to by the affected groups and individuals.

Finally each key area has five common features or attributes :

1 - commitment to perform and describe the actions needed to ensure that the process is established and will endure and typically involves policies and senior management sponsorship.

2 - Ability to perform and describe the preconditions that must exist in the project or organization to implement software process competently. Ability to perform typically involves resources, organizational sttructures, and training.

3 - Activities to performe and describe the roles and procedures necessary to implement a key process area. These typically involve established plans and procedures performing the work, tracking it, and taking corrective actions as necessary.

4 - Measurement and analysis which describe the need to measure the process and analyze the measurements.

5 - Verifying implementation describes the steps needed to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.

The intent of CMM is to describe needs to be done to develop and maintain software reliable as well, not how to do it. CMM further describes practices that con-

Thus, "CMM" is seen as a maturity or growth model in which a firm works its way up the five levels and, even after having attained level 5, is still in the process of continually improving and maturing.

Each of the five levels is also defined by the key processes associated with it. There are 18 key process areas that make up the five levels (see fig. 2: above). These processes are to be chosen because of their effectiveness in improving an organization's software process capability. They are considered to be requirements for achieving a maturity model.

Managerial processes are those that primarily affect the way operates to make decisions and control the project.

Technical processes are those tools that primarily affect the way developers operate to perform the technical software work.

Some software development firms and organizations. particularly in the U.S. started to use the capability maturity model to assess their software development process and software management. The importance of such assessment is to indicate the maturity and tecnological levels at which a firm is operating. More importantly, it may indicate the strong and weak areas of the firm's software development capabilities, thus identifying immediate imporvement priorities, interim improvement goals and progress measures.

One implication of the maturity process model for the software procedures in Egypt, for example is clear i.e., technology and managing the process of software engineering tangible products will become increasingly impotant. An understanding of the maturity model and how it might be of practical benefit to software development firms and organizations in Egypt, should be a research priority to be conducted.

II. GOALS OF THE CMM's KEY PROCESS AREAS :

The CMM provides a structure for each of the key process areas. Also, each key process area has one more than goals that are considered important for enhancing process capability.

The goals for the three of the key process areas in level 2, are shown in the following:

Level 2 : Repeatable. basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier success on projects with similar applications.

Figure 2: The fife levels and key process areas
of the capability maturity model

Level 5
Optimizing (Continually Improving Process)
* Process Change Management
  # Technology Change Management
  # Defect Prevention

Level 4
Managed (Predictable Process)
* Software Quality Management
* Quantitative Process Management

Level 3
Defined (Standards, Constituent Process)
  # Peer Reviews
  # Software Product Engineering
* Inter group Coordination
* Integrated Software Management
* Organization Process Defintion
* Organization Process Focus

Level 2
Repeatable (Discipline Process)
  # Software Configuration Management
  # Software Quality Assurance
* Software Subcontract Management
* Software Project Tracking and oversight
* Requirements Management

Level 1
Initial (Ad Hoc, Chaotic)
* Managerial Processes
  # Technical Processes

sign and code reviews, training programs, and increased organizational focus on software engineering, including measuring specific tasks in the development process. Some uncertainties remain about the value of the measurements, the best ones to use and the appropriate response to reviews. In this level, the software engineering practices that lead to consistent output are known and understood, and are used across the whole firm or organization.

(4) The Managed Process Level :

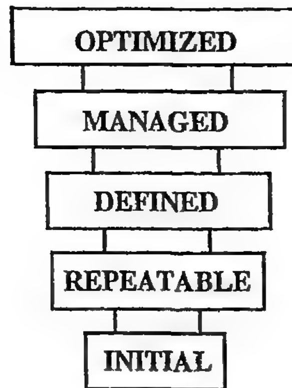
A firm typically has a minimum set of process management for each level in the software development life cycle, and it conducts extensive analysis of the data gathered during reviews and tests. Automated tools and techniques are used increasingly to control and manage the development process, as well as to support data collection and analysis. Therefore, within tthis level, the defined elements from the proceding level No. 3 are quantitatively instrumented, so that the level No. 5, the optimization level can be achieved.

(5) The Optimized Process Level :

This level exists in firms and organizations in which their development process operates smoothly as a matter of routine and continuous process improvement is conducted on the defined and quantified processes established in the previous levels. Therefore, these firms at this tevel have achieved a high degree of control over their process, have automated data gathering and typically have a method for improving and optimizing these operations. This includes identifying and replacing obsolete technologies, more sophisticated analysis of error and test data, and the introduction of error cause analysis and preventive studies [3].

The following figure (2) illsutrates these five levels of the "CMM". The items listed are called key process areas. These areas determine an organization's software development maturity.

Figure 1: Software Development Maturity Model



This model is utilized if firms and organizations interested in software development can judge the effectiveness of their software process has five maturity levels : initial, repeatable defined, managed and optimized.

(1) The Initial Process Level :

A software development firm has ill-defined procedures and controls. It typically, operates without project controls and does not integrate tools and techniques with the process. Coding and testing are being dominant activities. Established procedures, if they exist, are usually abandoned in a crisis. Therefore, This level is described as ad hoc, poorly controlled, and often with unpredictable results in terms of schedule, effort, and quality.

(2) The Repeatable Process Level :

A Software development firm or organization established basic project controls, such as : project scheduling, coding standards, software quality assurance, and change control. Mechanisms are in place for ensuring that the designer or the design team understands each software requirement. Statistics may be gathered on software code and test errors. The strength of the software development firm may stem from its prior experience at doing similar work, but it may face major risks when presented with new techniques. Therefore, in this level, the outputs of the process are consistent (in terms of schedule, effort and quality) and basic controls are in place, but processes that produce these results are most defined and understood.

(3) The Defined Process Level :

Standards and methods for technical and management activities required for software development are established at this level. These specifically include de-

SOFTWARE DEVELOPMENT PROCESS

A major promise in software development process is that the quality of a piece of software which is governed largely by the quality of the process used to develop and maintain it.

Continuous support for a software development improvement effort requires at least tow things : a clearly defined improvement model to follow, and success at applying the model in the firm or organization.

Manufacturing enterprises are always looking for more efficient ways of producing products because they realize that an efficient process yields lower costs, better quality and increase customer satisfaction. Software developers and producers are no different from their hardware counterparts in that they want to use the best software development process available.

In any mature software process, the methods, techniques and technology are to be used effectively and produce reasonably consistent results. Improvements in quality and productivity occur in part from automation. But in an immature software developed process, unpredictable results occur : formal procedures, cost estimates and project plans are lacking; and technology is used on ad hoc basis;. The prospects of software development in Egypt will depend largely and increasingly on the processes carried out in the software development life cycle to be followed. There is considerable middle ground between these two development processes extremes. The Capability Maturity Model (CMM) for software developed by the Software Engineering Institute at Carnegie - Mellon University is a process model that provides excellent guidance to improve software development processes.

I. THE MATURITY MODEL :

The Capability Maturity Model (CMM) [2] is used to evaluate and improve the way software is built and mentained. This model was first released in 1987, it was originally based on the experience of members of the Software Engineering Institute. The "CMM" has been continuously improved and refined since 1987 through successive revisions based on industry - wide and world - wide input. Yet, even though it is based on experience, it is only a model, which is an abstract, general framework describing the process used to develop and maintain software.

Like any model, it requires interpretation to be used in specific context. The approach used by "CMM" is to describe the principles and leave their implementation up to the managers and technical staff of each firm which will tailor "CMM" according to the culture and experiences of its own environment.

Perhaps the most well - known aspect of the CMM is its description of fife stages or maturity levels of a firm's software development process. These levels are shown in the following figure.

Without establishing processes, methods and techniques, software development firms and organizations often solve the same problems over and over again. The average large software project continuous to cost twice as much as its initial budget and it is completed a year or two behind its schedule; approximately 25% of such projects are never completed and remaining about 75% require inordinate amount of maintenance.

Therefore, the adaptation of new technologies and the more efficient computer programs are crucial for software industry. It is claimed that the market directs the evolution of software industry. The new market dominated nowadays, by the ***Internet*** will be important in short - term, which Egypt has to consider in its efforts to develop its software industrial sector.

The utilization of new technologies in software development and engineering is considered the basic guideline to have a share in the interantional market of software products. Therefore, a strategy of software industry should be underlined and emphasized by the concerned parties. There are two advantages of such a strategy:

* the first one, should bring the software development firms closer to customers or users, who, all over the world are not looking for one time transaction. They want at least a ten year partnership. This model helps to understand the challenges that firms and organizations are going through.

* The second software strategy, deals with by breaking up into industry units, we can maintain our entrepreneurial culture and continue to grow like a small company or firm. Therefore, a core software development center should be maintained for the firm's human resources and financial module. but responsibility for all sales and services will rest with the individual business units.

Taking, these two strategies in consideration, software products and new technologies should be patented for preserving intellectual property rights. For example, the U. S. Patent and Trademark Office (PTO) issued about five thousands patents for software products in 1993 [1]. This is an eightfold increase over the previous decade. The number was expected to exceed 5500 with the year 1995. It is stated that the most popular areas of software patents are; image processing, followed by network and communication systems. Anecdotal evidence suggests that U. S. and Japan hold half of all the software patents issued world wide.

(4) An expansion and fragmentation of industry resulting in large number of independent software vendors.

The technology of software development as related to tools and methods, as defined broadly as procedural methods, organizational models, and technological knowledge of development tools which are used to transform inputs into outputs, is becoming an increasingly important element in international competitiveness of organizations and management of software production process of firms.

This paper reviews some new software development practices being introduced by software development firms to assist in software engineering and software control costs and improve the quality of both the final products and the process of developing software. Some new technologies that will affect the entire software production process and that will require a reorientation in thinking about the investment and technological options are examined in this paper.

Therefore, the paper highlights the Capability Maturity Model (CMM) which gives software development firms and organizations a framework for evaluating their development processes and taking appropriate steps to improve them. The model's five levels include the initial level at which business adopt the program. In the second level, software development firms are able to deliver their software products on a consistent schedule and with a specific quality level. In the defined level, the developers begin to understand what practices let them to deliver consistently, and they disseminate the practices across their firms. At the management level, the practices are quantified, and at the optimized level, the new programs are established as routines and maintained with constant improvments.

INTRODUCTION

Software development firms and organizations in Egypt as in other developing countries are still struggling with inappropriate tools and methods utilized in software development process. This problem is due to the following factors :

(1) Lack of knowledge on the discipline.

(2) Lack of experience.

(3) Difficulty of measuring the development effort accurately.

(4) Implementation of designs with which poor quality does not surface until the finished products is either tested or installed for operation.

(5) High life cycle costs resulting from a system that was not designed for reusability or maintainability.

CHAPTER 6

TECHNOLOGICAL IMPLICATIONS ON SOFTWARE DEVELOPMENT

Porf. Dr. Mohamed M. El Hadi

ABSTRACT

The World - wide computer industry has been characterized since its inception by continuous innovation, improvement and rapid change. The developments have resulted in the growing importance of software, i. e. the programs that run computers and allow them to communicate with each other through data network.

Egypt as a newly developed country is trying to seek a share in the on-going global information of its software industry through a variety of policy and industrial measures. At the same time, a growing number of software development firms and organizations are beginning to develop software to be marketed domestically and abroad.

The world - wide software industry is still very much new mental industry. Any assessment of international software industry and market has to rely on insufficient, inconsistent and unreliable data. This makes the projections for the future software products and services an exceedingly risky undertaking. Nevertheless, such projections have been made and the range from 70 - 180 billion US $ by 1990 which has doubled by now.

There are at least four major forces which promoting rapid changes of software industry in the world at large, and in the Egyptian situation in particular; These forces are as follows:

(1) A productivity bottleneck in programming, with software and software - related support activities now accounting for the overwhelming of total system costs.

(2) The move away from single vendor solutions : as the typical way for organizations to meet their information systems needs towards customized, integrated, multi - vendor hardware and software solutions.

(3) The increasing emphasis on software production and sales by hardware vendors, leading construction by large and medium sized firms.

[18] K. H. Kim, *"Software Fault Tolerance"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

[19] M. L. Gabri, *"Total Quality Management"*, Hanns Seidel Foundation, Management Traning Project, Cairo, 1993.

[20] M. M. El-Boraey, *"Reliability of Parallel Processing Systems"*, Ph. D. Dissertation, Computer & Systems Eng., Al-Azhar Univ., 1992.

[21] M. P. Mariani, & D. F. Palmer, *"Software Development for Distrbuted Computing Systems"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[22] Michael J. Lutz, "Consumable Mathematics for Software Engineering", IEEE COMPUTER SOCIETY, APRIL 1996.

[23] R. D. Williams, *"Management of Software Development"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

[24] R. W. Wolverton, *"Software Costing"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

[25] S. H. Saib, *"Formal Verification"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

[26] Thomas Drake, *"Measuring Software Quality : A Case Study"*, IEEE COMPUTER SOCIETY, NOVEMBER 1996.

[4] B. W. Boehm, *"Software Life Cycle Factors"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[5] Bernard P. Zeiglar, *"Theory and Application of Modeling and Simulation : A Software Engineering Perspective"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[6] Boris Beizer, *"Software Performance"*. HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[7] Boris Beizer, *"Software Systems Testing and Quality Assurance"*, Van Nostrand Reinhold Company Inc, 1984.

[8] C. R. Vick, *"Introduction : A Software Engineering Environment"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[9] Daniel Jackson, and Jeannette, *"Lightweight Formal Methods"* IEEE COMPUTER SOCIETY, APRIL 1996.

[10] Don Oxley, Bill Suber, & Merrill Cornish, *"Software Development for Data Flow Machines"*. HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[11] Edward F. Miller, *"Software Testing Tehnology : An overview"*, HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc, 1984.

[12] Fathey E. Essa, *"The use of Temporal Logic in Software Testing"*, Ph. D. Dissertation, Computer & Systems Eng., Al-Azhar Univ., 1988.

[13] Gregory A. Hansen, *"Simulating Software Development Process"*, IEEE COMPUTER SOCIETY, JANUARY 1996.

[14] Hon H. So, *"Graph Theoretic and Analysis in Software Engineering"*, HANDBOOK of SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

[15] Castro, *"Distrbution System Specification using A Temporal-Causal Framework"*, Ph. D. Dissertation, London Univ., October 1990.

[16] John D. Musa, *"Software Reliability-Engineered Testing"*, IEEE COMPUTER SOCIETY, NOVEMBER 1996.

[17] John D. Musa, *"Software Reliability"* HANDBOOK OF SOFTWARE ENGINEERING, Van Nostrand Reinhold Company Inc., 1984.

6. CONCLUSION AND FUTURE WORK

The main goals of this paper are :

- ♦ to introduce the concept of formal incremental specification,
- ♦ to present a quality assurance model for software development process, and
- ♦ to improve the increment development of software products by refining the specification and conducting the quality assurance model to all the development phases.

In this paper software engineering, and quality assurance tools and techniques are considered the foundation of the development of software products improvement.

The formal incremental specifications means that the specifcation must be classified into increments : cause-effect specification increment, real time specification increment, temporal specification increment, and structural specification increment.

The incremental specification is considered a comprehensive specification of the software product as the base of the development process. Applying the quality assurance model to all the software development phases represent the foundation to the software product total quality.

As one of the parallel processing machines; data flow machine is considered as a case study to apply the formal incremental specification and the quality assurance model. The specification and quality assurance tools and methodologies for the data flow machine are presented.

Much more efforts must be done to define new quality assurance factors and their testing methodologies for the tools of each of the specification perspectives.

Much more effort must be done to produce automated tools for the formal incremental specification and quality assurance specifically and the development life cycle generally.

REFERENCES AND BIBLIOGRAPHY

[1] Ahmad Taha Shehab El-Dean, ***"Hardware Testing Temproral Logic"***, Ph. D. Dissertation, Computer & Systems Eng., Al-Azhar Univ., March 1997.

[2] Alan Wood, ***"Predicting Software Reliability"***, IEEE COMPUTER SOCIETY, NOVEMBER 1996.

[3] Albert Y. Zomaya, Editor, ***"Parallel and Distributed Computing Handbook"***, McGraw-Hill 1996.

5. INCREMENTAL DEVELOPMENT FOR DATA FLOW MACHINE SOFTWARE

The development process of data flow software products can be improved by adhering the incremental model of software development. The main problem is the extraction of parallelism [10].

There are three levels of parallelisms :

♦ Program level : independent programs are executed in parallel,

♦ Task level : programmer specified procedures (tasks) are executed in parallel, and

♦ Instruction level : independent operations within an expression are executed in parallel, independent serial sequences are executed in parallel, and independent functions are executed in parallel.

Accoding to the complexity of the structure and the behavior of the data flow systems; there is a need for the incremental specification to specify all the levels of parallelism accurately and completely.

Applying the Incremental Specification [1. 12, 20, 21]

♦ **Cause-effect specification** : the interaction among the components of the system (at each level of parallelism) is specified using **Petri Net** as a formal specification tool. The performance quality assurance factors are the safeness and liveness. The reachability analysis is used as a testing methodology to asses the quality assurance factors.

♦ **Time specification** : the amount of time that can be taken by each of the interacting components is specified using the **PERT chart** as a formal specification tool. The performance quality assurance factor is the minmum required time for the interacting components to terminate. The Critical Path Method (CPM) is used as a testing methodology to asses the time requirements. Also Timed Petri Nets (TPN) can be used to specify the real time interaction among concurrent processes.

♦ **Temporal specification** : The temporal relations among the interacting processes are specified using Temporal Logic (TL). The performance quality assurance factors are the availability, safeness, and liveness. The theorem proving methodology is used to asses the quality assurance factors.

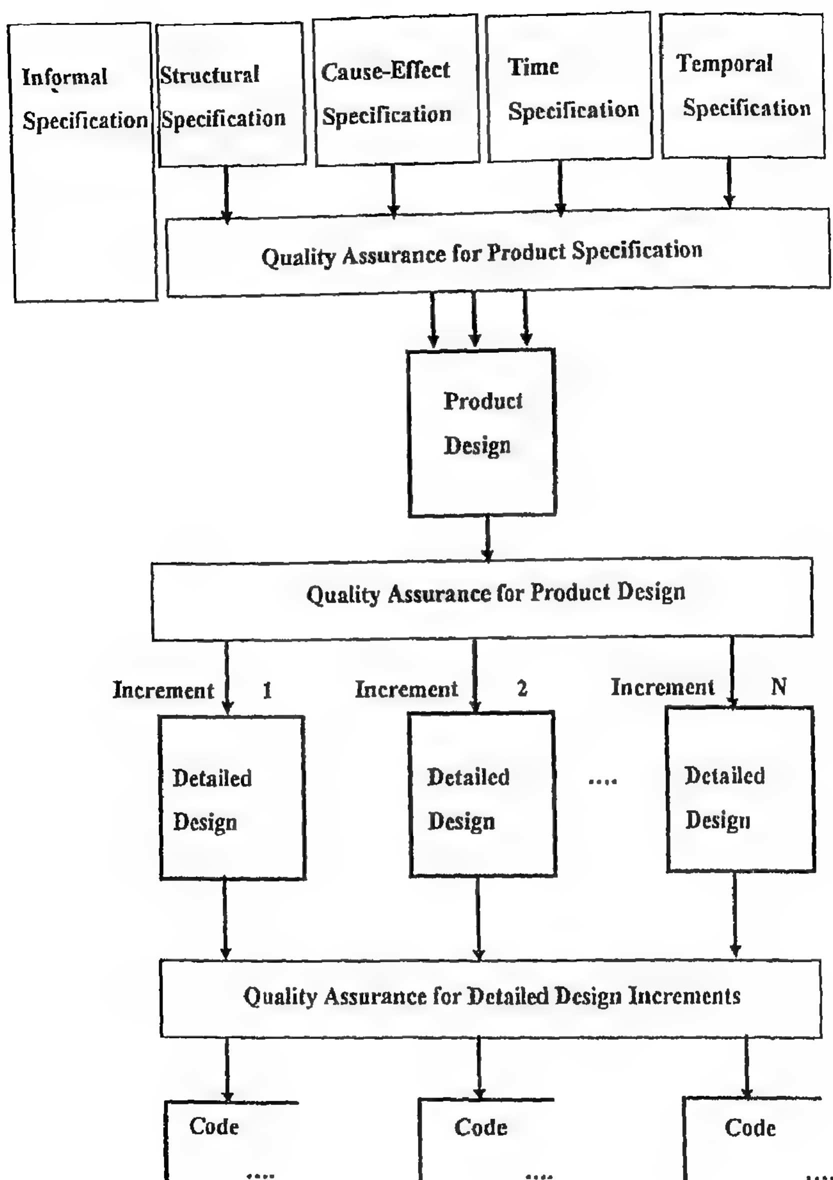


Fig. 4.3 : The complete incremental model of software development.

♦ **The Incremental Formal Specification of Software Products :**

The incremental development as shown in fig. 4.2 considers the detailed design phase of the development as the base to start the increments of the development process.

In fact the specification is the basis of development, also it provides the communication between the user and the developer and it often serves as a standard document across the remainder phases of development. The specification phase of the development process is very important since the specification errors, if not fixed very early, need a great effort and cost to fix [6, 7, 11, 15, 16, 17, 18].

Today software specification is considered as an independent area of study because of its importance to the engineering discipline. Specification leads to proposals for simplification, which in turn lead to a system that is easier to design, implement, and maintain [15].

Formal methods for specification provide the bases of specification quality assurance, and so increasing the confidence in the developed software [15, 22]. Unfortunately, most of software industries use informal methods for specification. Informal specifications may be easy to read and understand, but they are often incomplete, vague, ambiguous, and can not be quality assured. Formal specification must take into consideration the structure and the behavior of the specified system [1].

Formal behavioral specifications consider three perspectives:

♦ cause-effect specification,

♦ time specification, and

♦ temporal specification.

Each of the three perspectives has its tools and techniques to specify systems and to define the quality factors for that specification.

Complete behavioral specification must posses the three behavioral specification perspectives. The incremental specification process must consider the three perspectives as formal increments of the behavioral specifcation as well as the structural specification and the informal behavioral specifications. The informal specification is needed as explanations for the formal specification. The complete incremental model of software development is shown in fig. 4.3.

incremental development means that the software is developed in increments of its functional capability. The advantages of incremental development are the testability of increments and the cost effectiveness of product refinement. The Waterfall model using the incremental development is shown in fig. 4.2.

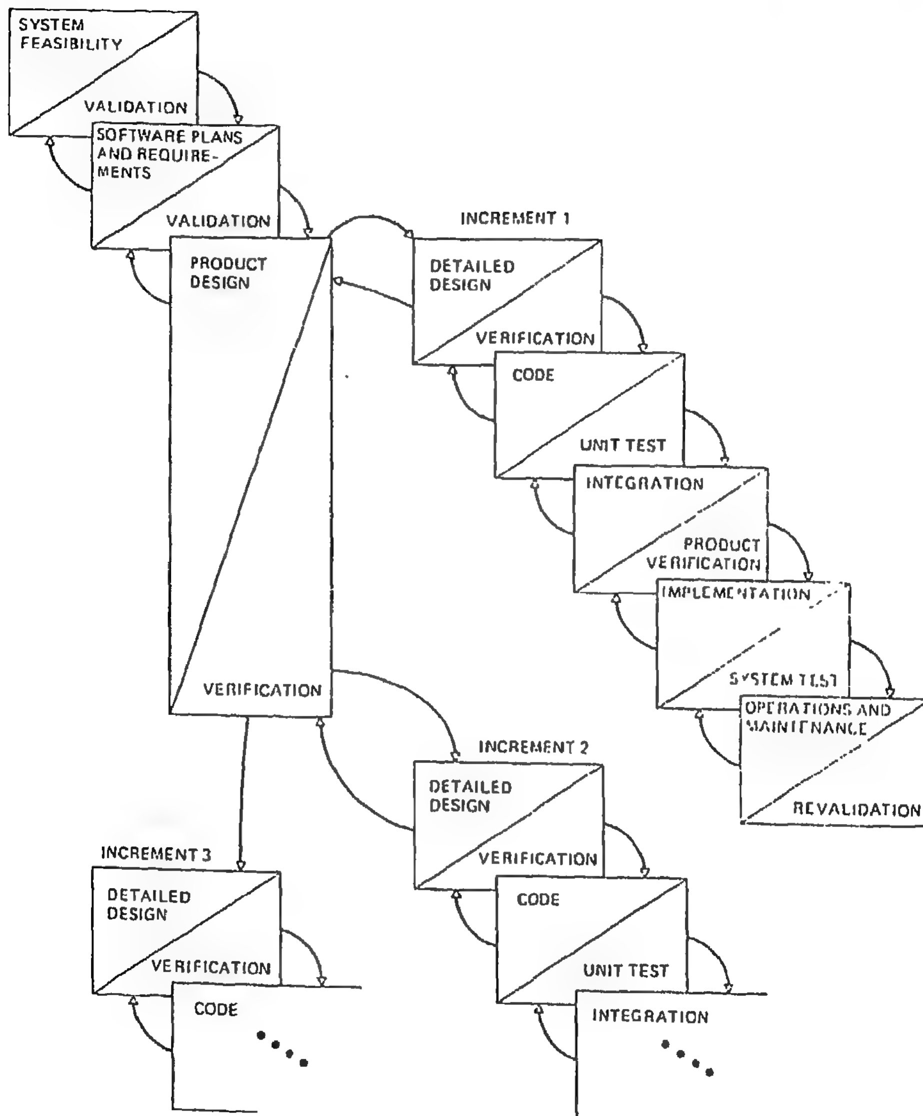


Fig. 4.2 : The Waterfall model using the incremental development [4].

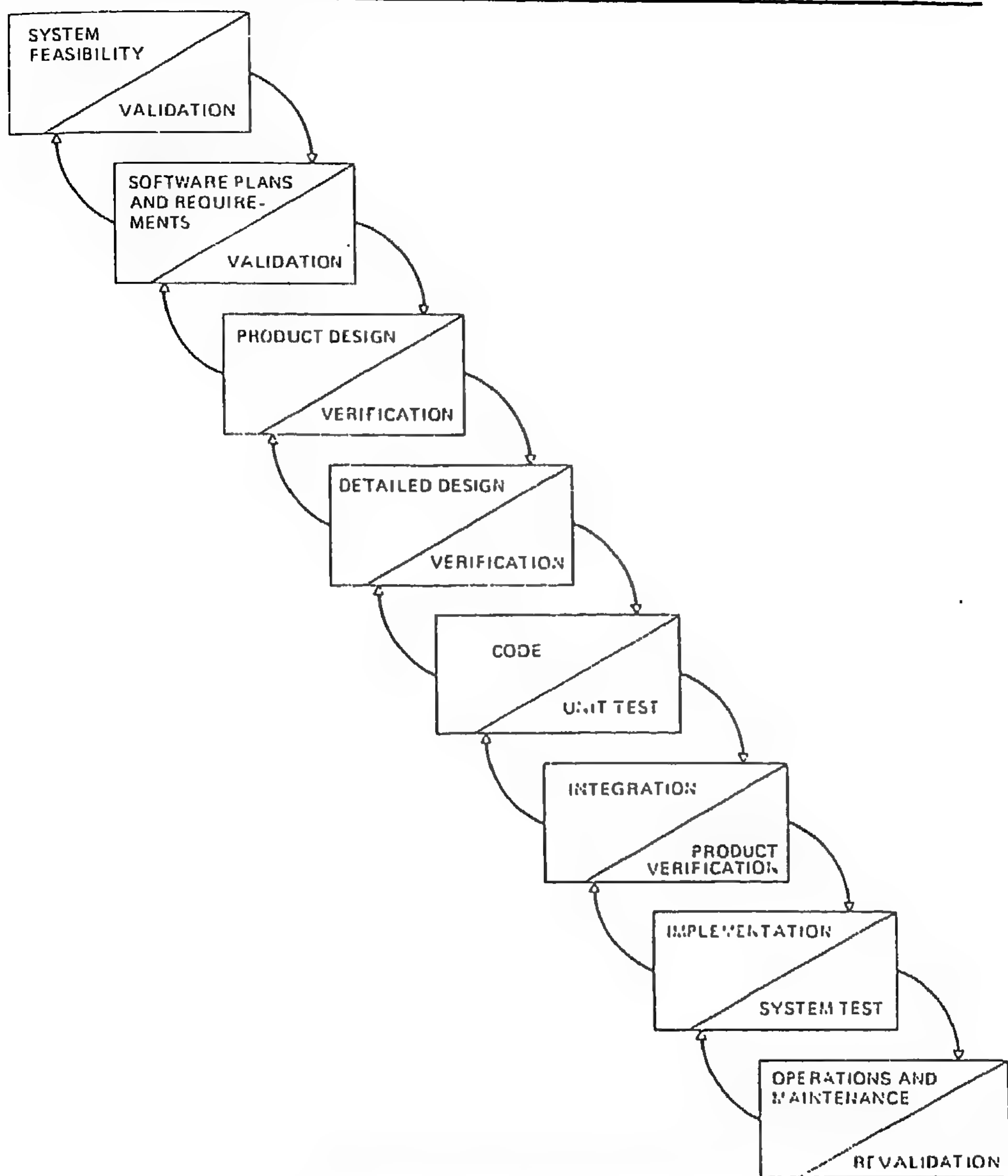


Fig. 4.1 : The "Waterfall" model of the software life cysle [4].

In fact it is cost-effective to modify the sequence of software life cycle. Incremental development of software product is one of the most important Waterfall refinements.

B.W. Boehm introduced a refinement to the Waterfall model of software development process; that is the incremental development of software products [4]. The

correcting the deviations from the requirements and measuring the software quality factors [2, 19, 26].

Testing is the primary tool of software quality assurance. Quality assurance concerns not only performing a test, but also by defining the software products quality factors, establishing standards for test, and introducing methodologies for test design and test cases generation [7, 16, 17, 26].

Software quality assurance must be conducted; in a specific methodology, to each phase of the software development process as well as to the final software product to achieve the total quality of the software products [19, 26].

Software quality assurance holds great promise for improving the practice and moving the profession along the path toward a full software production engineering discipline [16, 17].

A proposed general model for quatlity assurance is shown in fig. 3.1

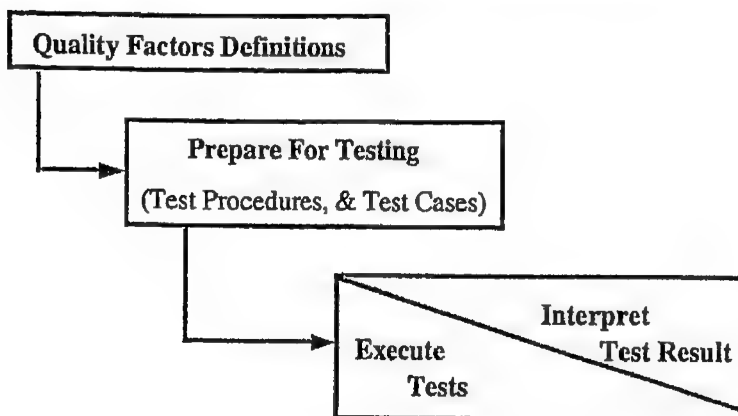


Fig. 3.1 : The proposed quality assurance model.

4. THE INCREMENTAL DEVELOPMENT OF SOFTWARE PRODUCTS

The development process of software products is an engineering activity; so it is classified into a set of phases [22]. The waterfall model; shown in fig 4.1; presents the software products life cycle.

3. Software Engineering and Quality Assurance

3.1 Software engineering :

Software engineering is a discipline concerned with the production and maintenace of software products. As other engineering fields, software engineering concerns of the development of software products in an effective, economical, and timely fashion. Software engineering is based on the foundation of computer science, management science, and engineering approach to problem solving.

The main goal of software engineering is to achieve the right balance among quality, productivity, and cost effectiveness of software product [6, 7, 8, 16, 17, 21].

3.2 Software Quality :

There are many definitions for the quality of products generally, and the quality of software products specifically. The following are some quality defintions [19]:

♦ **From the manufacturing point of wiew;** the quality means conformance to requirements

♦ **From the usability point of wiew;** the quality means the intensity of failure (reliability).

♦ **From the product value point of wiew;** the quality means the degree of excellence at an acceptable price.

♦ **From the customer point of wiew;** the quality is the capacity to satisfy wants.

♦ **American National Stanard Institute;** the quality is the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs.

The definition of software quality can be applied in the development process of the software product as in defintion 1; as well as; in the operation phase as in the definitions 2, & 4.

The most serious problems facing the software industry are the low-quality problems.

The Software Engineering Applied Technology Center (**SEATC**) for the National Security Agency (NSA) addresses software quality as a key initiative to imporove software development [26].

3.3 Software Quality Assurance (SQA) :

Software quality assurance is the methodologies and tools of preventing or/and

MIT Static data flow machine consists of four units:

- ♦ *Memory unit* : it is a collection of blocks; cell blocks; of central storage locations. Each block stores a node information which are operation, operands, and the destination addresses. The fist step to execute a program graph is to load it into the cell blocks. A part of the memory is used to store arrays to support rapid access of array elements.
- ♦ ***Processing section*** : it comprises eight identical processors. The instruction set contains floating point, fixed point, logical packet communication, and shift instructions.
- ♦ ***Arbitration network*** : it communicates the enabled packet of instructions from the cell blocks of the memory to the processing elements using a **FIFO** buffer.
- ♦ ***Distribution network*** : it communicates the results of execution from the processing elements to the cell blocks of the memory using a **FIFO** buffer.

MIT static data flow machine has one communication path of 32 bits width and 11 words instruction packets. If a ***processor is faulty,*** the firing node is lost, i.e. no result will be conducted to the next node or other processors, so the computation is restarted from the beginning [21].

The execution of a data flow language program has the following steps:

- ♦ ***Data Flow Graph Construction***
- ♦ ***Parallelism extraction***: instructions that are available for concurrent execution are extracted.
- ♦ ***Program execution:*** the execution of a program is the firing of all of its data flow graph nodes. For each node to be executed the following steps are required:
  - • ***Node Reading*** : The node information is read from the memory.
  - • ***Node Enabling*** : if all of the node input tokens are available, the node is enabled to be fired.
  - • ***Node Firing*** : the node is aassigned to an available processor to execute the operation associated with that node using the functional units of the processor; then the resultant token is passed into the output arc.
  - • ***Result Distribution*** : the result is moved to the next successor nodes.

2.1 Data Flow Versus Contral Flow :

Data flow computers are based on the concept of data-driven computation that is different from the operation of the conventional Von-Neumann machine.

The main difference is that the instruction execution in Von-Numann is under program-flow control, while in a data flow computer is driven by data availabilty.

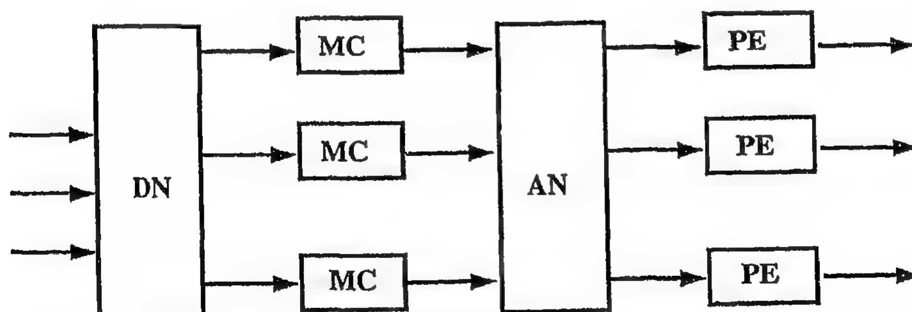
Data flow is a technique for specifying computation in a two-dimensional graphical form: instructions that are available for concurrent execution are written alongside one another, and instructions that must be executed in sequence are written one under the other. Information items in data flow computers appear as operation packets and data tokens. An operation is composed of the opcode, operands, and destination of its sucessor instructions. A data token is formed with a result value and its destination.

2.2 Data Flow Machine : [10, 21]

Data folw machine is one of the most promising parallel processing machines. Research and development on data flow computing have been started at 1968. There are two main classifications for data flow machines: Static and dynamic data flow machines [21].

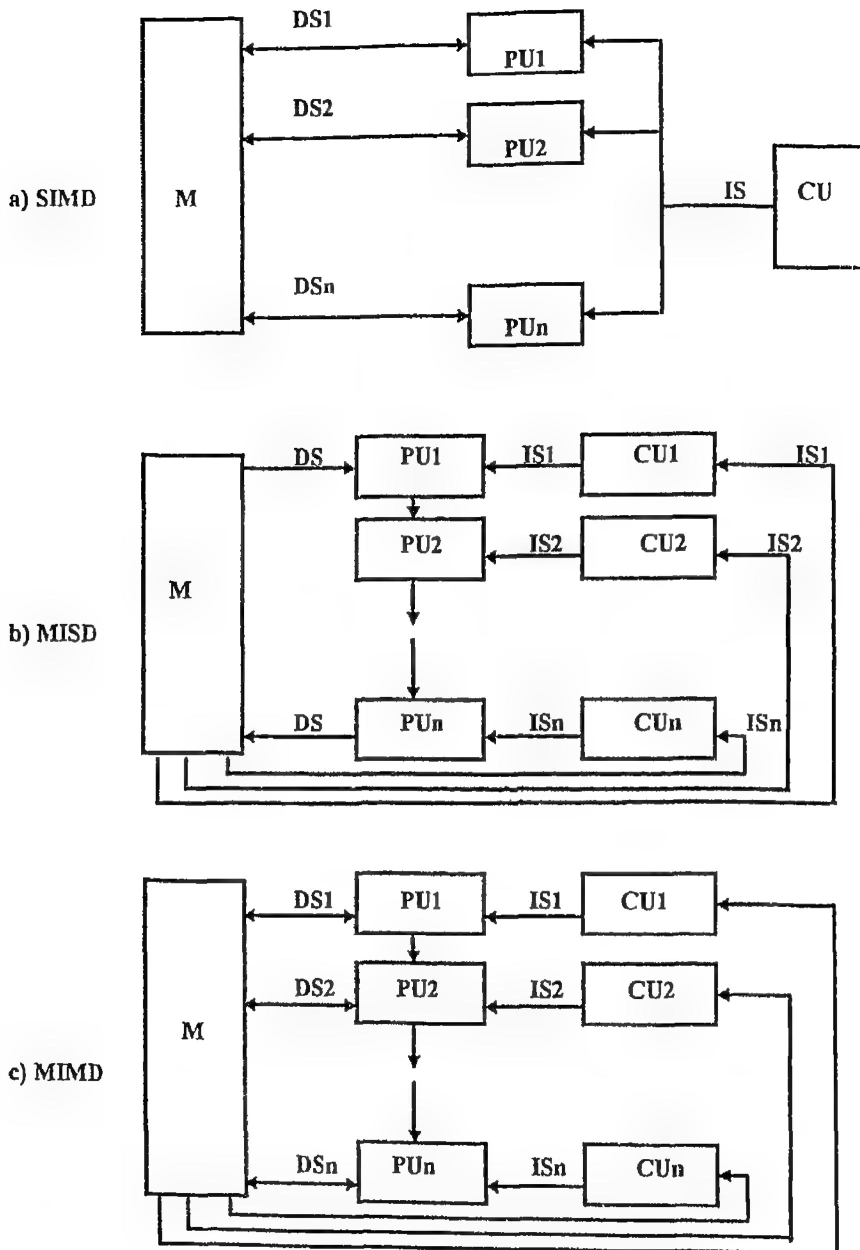
MIT Static Data Flow Machine

One of the static data flow machines is MIT machine. The block diagram of the MIT static data flow machine architecture is shown in fig 2.3.



DN : Distribution Network, MC: Memory Cell, AN: Arbitration Network, &
PE: Processing Element

Fig. 2.3 The architecture of the MIT data flow machine [21].



M: Memory, PU· Processing Unit, CU. Control Unit,

DS. Data Stream, & IS: Instruction Stream.

Fig. 2.2 Parallel processing architectures [3].

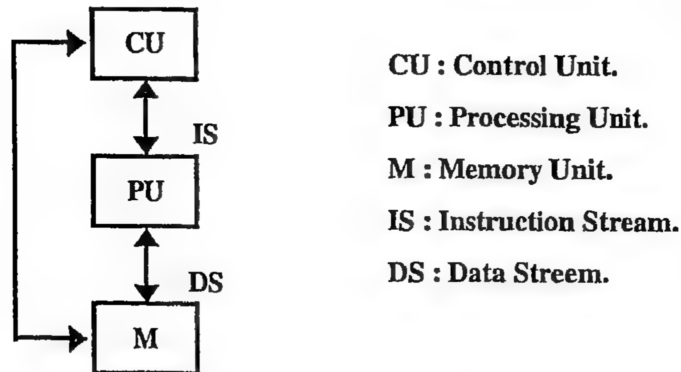


Fig 2.1 : The Von-Neumann Computing Paradigm [3].

Computers operate by executing instructions on data. The computer architecture is classified on the basis of how the machine relates the instruction stream to the data stream into 4 classes:

1 - Single Instruction stream, Single Data stream (SISD) Von- Neumann machines),

2 - Single Instruction stream, Multiple Data stream (SIMD),

3 - Multiple Instruction stream, Single Data stream (MISD),

4 - Multiple Instruction stream, Multiple Data stream (MIMD).

The architectures of SIMD, MISD, and MIMD are shown in fig. 2.2 (a), (b), & (c).

production of faster, more reliable, and cheaper electronic components; made it evident that the performance of uniprocessor computers is limited. The alternative to sequential processing is parallel procèssing machines. The transition from sequential to parallel and distributed computing offers high computing power that is very important to solve computationlly intensive problems [3,21].

Data flow machine is one of the most promising parallel processing machines. Conisderable progress is made in building data flow machines during the last few years. The data flow computing concept is the most effective, promising computing method to implement in machine architecture for high speed computing[21].

1.1 Motivation :

The production process of software products differs from the production of goods. The manufacturing costs of software products are neglected; it is the cost of copying a piece of software. Also, the maintenance process differs form software products to hardware products. In software products the maintenance process consists of two activities: correction and update. The corrective activity leaves the functional specification intact while the update acivity changes the functional specification of the software product. Really the maintenance process affects the early phases of the development process.

As the architecture of hardware becomes more and more complex; the supporting software products become more and more complex. As software products become more and more complex; the development process becomes more and more complex. As the development process of software product becomes more and more complex; software engineering and quality assurance tools & techniques become more and more vital to achieve cost-effective high quality product.

2. PARALLEL AND DISTRIBUTED PROCESSING PARADIGMS :

The majority of today's computers are conceptually very similar. Their architectures and modes of operation follow the basic design principles fromulated by John Von-Neumann and coworker [3]. The Von-Neumann approach is based on the simple idea of a control unit that fetches an instruction and its operands from a memory unit and sends them to a processing unit in which the instruction is executed and the result is sent to the memory. The Von-Neumann paradigm is called Single Instruction Stream Single Data steam (SISD) as shown fig. 2.1.

CHAPTER 5

Towards Improvement of software Products for Parallel and Distributed Systems

Dr. Ahmed Taha Shehab El-Dean

&

Dr. Mohammed M. Eassa

ABSTRACT

Information technology is advancing at enormous rate, but most of software products are facing some problems. Due to the compexity of parallel/distributed systems; the development of their software products is affected by many factors and so it is eroor-prone and costly.

In order to achieve cost-effective with high quality; parallel/distributed software products; much more efforts must be done in all the phases of the development process specially the very beginning ones; the specification and design phases.

Software engineering and quality assurance techniques must be the foundation of the production of parallel/distrbuted software systems.

This paper introduces a framework that combines the software engineering and quality assurance techniques to improve software products for parallel/distrbuted systems. The proposed framework for the development of software products is applied on the data flow machine as a case study.

1 - INTRODUCTION

Computer hardware is improved from vacuum tubes to VLSI, but there has been no significant change in the sequential abstract computing model. The demand for ultrahigh speed computing machines is incrasing every day. The major difficulty in satisfying this demand in uniprocessing is the physical constrains of hardware and the sequential and centralized control in the Von-Neumann computing model. The

21. K. S. Shankar; "A Functional Approach to Module Verification", IEEE Transactions on Software Engineering, Vol. Se-8, No. 2, February, 1982, pp.147-160.

22. K. S. Shankar; "Data Structures, Types, and Abstractions", IEEE Computer, April, 1980, pp. 67-77.

23. H. D. Mills, R.C. Linger, and A.R. Hevner' "Box information Systems", IBM Systems Jaurnal, Vol. 62, No. 4, 1987, pp. 395-413.

24. H. D. Mills' "A Stepwise Refinement and Verification in Box Structured Systems", IEEE Computer, June, 1988, pp. 23-36.

25. M. S. Deck' "Cleanroom and Object- Oriented Software Engineering: A Unique Synergy", Proc, 8th Annual Software Technology Conference, Salt Lake City, UT, USA, April, 1996.

26. M. D. Deck; "Cleanroom Review Techiques for Application Development", Proc. 6th International Conference on software quality Ottawa, Canada, October 1996.

27. A. R. Hevner, "Object-Oriented System Development Methods", Advances in Computer, Vol. 35, Academic Press, 1992, pp. 135-148.

28. R. S. Pressman; "Software Engineeing: A Parctitoner's Approach, 3rd Edition, MCGraw-Hill, 1992, pp. 180-183.

29. W. E. Eureka and N. E. Ryan, eds.; "Quality Up, Costs Down: A Manager's Guide to Taguchi Methods and QFD", Irwin Professional Publishing, Burr Ridge, IL, USA,, 1995.

30. Y. C. Wong: "A Functional Semantics of Pointer Operations and its Application", IBM Software Engineering ITL, March, 1992.

31. M. D. Deck; "Cleanroom Software Engineering: Quality Imporvement and Cost Reduction", Proc. Pacific Northwest Software Quality Conference, October, 1994, pp. 243-258.

32. J. Rosen; "Design Languages for Cleanroom Software Engineering", Proc. 25th Hawaii International Conference on System Sciences, 1992.

33. B. Beizer; "Software Testing Techniques, 2nd ed., Van Nostrand Reinhold, 1990.

7. R. W. Selby, V. R. Basili, and F. T. Baker; "Cleanroom Software Development: An Empirical Evaluation", IEEE Transactions on Software Engineering. Vol. SE-13, No. 9, September, 1987, pp. 1027-1037.

8. J. Trammel, L.H. Binder, and C. E. Snyder; "The Automated Production Documentation System: A Case Study in Cleanroom Software Engineering", ACM Tranc. On Software Engineering and Methodology, Vol. 1, No. 1, January, 1992, pp. 81-94.

9. P. A. Hausler; "A Recent Cleanroom Success Story; The Redwing Project", Pro. 17th Annual Software Engineering Workshop, NASA Goddard Space Flight Center, December, 1992.

10. Jan Graham; "Object-Oriented Methods", Second Edition, Addison-Wesley, 1994, pp. 8-9.

11. G. Booch; "Object-Oriented Design with Applications", Redwood City, CA: Benjamin Cummings, 1991.

12. J. Rumbaugh, M. Blaha, and W. Premerlani; "Object-Oriented Modeling and Design, Englewood Cliffs, NJ: Prentice-Hall, 1991.

13. S. Shlaer, and S. J. Mellor; "Object-Lifecycles: Modeling the World in States", Englewood Cliffs, NJ: Yourdon Press, 1991.

14. Booch & Grady; "Object-oriented Analysis and Design with applications", Redwood City, CA: Benjamin- Cummings, 1991.

15. Shlaer, Sally and Steve Mellor; "Object-oriented Lifecycles, Modelling the World in States", Prentice - Hall, 1992.

16. Jacobson, Lvar, Magnus Christerson, M. Jonsson, and G. Overganrd; "Object-oriented Software Engineering", Addison- Wesley, 1992.

17. Richard C. Linger; "Cleanroom Software Engineering for Zero-Defect Software", Proc. 15th International Conference on Software Engineering, May, 1993.

18. Linger R. & Mills H.; "A Case Study in Cleanroom Software Engineering: The IBM COBOL Stucturing Facility, Proc. 12th International Computer Science and Applications Conference, October 1988.

19. H. D. Mills' "The New Math of Computer Programming", Communications of the ACM, Vol. 18, No. 1, January, 1979, pp. 43-48.

20. R. C. Linger, H. D. Mills, and B. I. Witt' "Stuctured Programming: Theory and Pracitce", Addison-Wesley, 1979.

sufficiently compete that the risk of proceeding is minimized"?.

However, this model is to be distingushed from "iterative hacking", where there is no clear documented goal of development, and the code- test- debug cycle proceeds as rapidly as possible until either the delivery date arrives or someone declares it to be "complete". Throughout the paper we have noted that there are some places where additional tools and research would be useful. These include:

- Tools to identify ripple effects of changes, especially those involving the class hierarchy.
- Documentation black box specifications for widely used application frameworks.
- Tools to manage the hierarchy of abstractions and perform automatic flattening.

While the absence of such tools is not a barrier to effective Cleanroom use with object-orientation the tools would enhance the productivity of teams that combine these approaches and would make the unique synergies between them even clearer.

Where Cleanroom techniques may be easily added to the stated OO methods, technique integration may depend on successful demonstration, cost benefit analysis, and the desire and will to improve software quality.

6. REFERENCES

1. H. D. Mills, M. Dyer, and R. C. Linger, "Cleanroom Software Engineering", IEEE Software, September, 1987, pp. 19-25.
2. R. H. Cobb, and H.D. Mills; "Engineering under Statistical Quality Control", IEEE Software, November, 1990, pp. 44-45.
3. M. Dyer, "The Cleanroom Approach to Quality Software Development", Wiley, 1992.
4. Kouchalcdjian; "lessons Learned Using Cleanroom Software in the Software Engineering Laboratory", MSc Thesis, University of Maryland, College Park, 1990.
5. M. Dyer, and H. D. Mills' "The Cleanroom Approach to Reliable Software Development", in Proc. Validation Methods Research for Fault- Tolerant Avionics and Systems Sub- Working- Group Meeting: Production of Reliable Fligh-Crucial Software, Research Triangle Institute, NC, Nov. 2-4, 1981.
6. P. A. Currit, M. Dyer, and H.D. Mills; "Certifying the Reliability of Software" IEEE Transactions on Software Engineering, Vol. SE, No. 1. January, 1986, pp. 3-11.

whose abstract attributes are part of a "formal" attribute set. Parameter substitution of the instance name for the formal attributes set would occur when the object is instantiated or used, just as one would do with a procedure call. The box structures approach provides for either a strong separation (e.g., stimulus history) between abstract and concrete attributes or a weak separation (e. g., integer ID for both). It also provides for a separation between abstract processes and their implementations. Every box has stimulus-response behavior, and a box always responds to one message before accepting the next. However, there are certain difficulties at the implementation level where, for example, an object may receive one message while still processing a prior one. Concurrency is also a complicating effect. In practice, these problems can be resolved by introducing a more robust or complex mathematical model, or by paying careful attention during review to issues like reentrancy and shared-data conflict resolution.

In an object-oriented system, objects are also related to one another by inheritance. From the point of view of box structures, inheritance is a way of contructing a new data abstraction without having to copy an existing one. Depending on how we define these specifications, a user of the derived class may or may not know it is a derived class. Different languages have different facilities for defining objects; we believe this specification approach can be used, with slight variations, in all cases. Polymorphism and genericity present a signifcant challenge to understanding object-oriented systems, because it may be diffcult to determine which method is being invoked. We have described two approaches to documenting these methods. One way is to completely document such a statement, using one conditional, concurrent assignment for each possible virtual method. Another technique would be to have each method be so similar at the abstract views were identical, even though their concrete implementations are differed. The latter is more in keeping with the *Intent* of polymorphism, but the writing of good, clear, reusable abstractions is diffcult. In some of the preceding sections it may have appeared as if the Cleanroom development is strictly top-down: the box specification is developed to ultimate completion and design begins only at that point. This may reprresent an ideal circumstance in some project situations, but it is far from normal. In practice, some amount of specification is done, then some design or architecture work, and then additional specification. Reviews are used to make sure the evolving design meets the evolving specification. and to make modifications to either or both when necessary to keep them synchronized. Much designing takes place before and during the specification activity for any object. This is an "elevator" model of development: One moves up and down in the hierarchy of objects, but the last ride is top-down, to force the developers to think primarily in terms of what the system does, instead of how the system does it. The test for when any phase is "done" is usually, "is it

time-to-market. The decision to step back from formal verification is a complex one involving management and technical staff. If done, cycle time may be only marginally improved since the cost of fixing defects found in testing will likely increase, as will the cost of technical support. Since the Cleanroom process is an incremental one, some amount of adjustment can be made throughout the process, but a clear understanding of the goals and priorities is important at the outset. Although this opinion is not universal in the Cleanroom community, it is this author's belief that acceptable Cleanroom team review techniques span a continum between informal inspection and full formal correctness proofs [31]. It is for each team to decide, on an object-by-object basis, what point on this continuum to choose based on the priorities and conditions at that time. An important part of that decision is an assessment of the risks engendered by choosing to de less.

To illustrate, let's again take the example in Figure 4 (the conditional integer exchange) and think about what would be required for a "full" verification. Before we de so, we must point out that the code itself is a model for the object, and thus our precise understanding of the language semantics are critical in program verification. Since achiving that level of understanding for the entire C ++ language can be very difficult, it may be useful to agree on a subset of C ++ or even to choose design language [32]. With that said, it is very much up to us what level of rigor we want to present in our abstractions, and what parts of the semantics we feel comfortable leaving for careful inspections or even for testing.

There are many techniques for demonstrating correctness, and considerable variation in the "standard of proof" required. To use a legal analogy, a team designing life-critical systems might ask for correctness to be demonstrated "beyond a reasonable doubt", whereas a team designing a prototype word processor might only demand that the weight of the evidence points toward correctness. It is the author's opinion that, for some projects and some objects, inspection techniques suffciently meet the requirements of a "Cleanroom" review. One of the major concerns with sliding the standard of proof away from full verification, is that Cleanroom testing is pressed into greater service as a way to find errors instead of focusing on measuring quality. It may be prudent, in some project environments, to back up the Cleanroom reviews with formal testing techniques [33] if less rigorous.

5. CONCLUSION

There is a relationship between box structures and object evientation. In an object-based system, each object (instance) is specified by a data abstraction. Each instance has its own "actual" set of abstract attributes to which a process abstraction would refer. In a class-based system, each class has an associated data specification

applications. For example, the Microsoft Foundation Classes (MFC) and Borland's Object Windows Library (OWL) are frameworks for creating programs to run under Windows. In the pre-frameworks days, programming for Windows required manipulating a large number of application programming interface (API) functions in order to even display a rudimentary window on the screen. Now, it is as simple as creating an instance of the TWindow class (in OWL) and sending it the "Show" message. Of course, the increase in power is somewhat offset by a decrease in flexibility, and in most framework there are ways to directly access the underlying Windows API. It is for the latter reason that most framework classes present an abstract model (e. g. window) that is nearly identical to its implementation.

There are two special considerations when using application frameworks with Cleanroom. First, the class inheritance hierarchy can grow to be quite deep (as much as 6 deep) within the framework itself, even before the application devlopment team has begun its work. There are dozens of classes and hundreds of methods, combining all types of polymorphism and virtual binding. Understanding the abstract model of each class and the actual function of each method - the black box behavior of each object - is a significant task.

There are many other challenges in framwork-based application development. One is that programs are typically "dense" in method invocations: perhaps half or more of the lines of code are method invocations. Another is that the framework are often packaged with application and class generators that alleviate much of the tedium of program construction. All of these considerations motivate careful and judicious use of intermediate specifications, with it being extremely unlikely that every method invocation will possess one. It also places a premium on development teams that are very familiar with the framework in which they are developing, so as to be able to treat many kinds of objects as if they were language primitives like integer and string.

Finally, it is worth noting that most frameworks require applications to use pointers and arrays, and other "less Cleanroom-desirable" constructs. Although techniques for full formal verification of pointer operations have been proposed [30], a more common approach is to always use care to make pointer-accessed data objects as similar to full-fledged objects as possible, and to verify them as you normally would.

4.6. Verifcation and Inspection: Choices and Guidance

Although most Cleanroom literature has emphasized the mathematical purity of verification, there are many projects that do not require life-critical reliability levels and are willing to accept lower reliability levels, if doing so will improve their

evaluating whether the specification will meet customer needs. Clearly this is an important aspect of system quality, yet there is nothing unique either Cleanroom or OOA that addresses it. Techniques such as prototyping (either rapid prototypes or early increments), facilitated application specification technique (FAST) [28], Quality Function Deployment [29], and through team review can all make important contributions. Many objects in the system will require a black box description, and some of them will be complex enough to warrant their own object-oriented analysis. In a pure world, such objects would not technically appear until the state box or clear box designs, and so their actual specification would begin at that point. However, if one chooses to have a close mapping between specification objects and design objects, then this division becomes much less clear, as does the entire distinction between the "specification phase" and "design phase."

4.4. Correctness Verification in OO Development

The heart of Cleanroom review is correctness verification based on the functional model. In that model, the overall function of a procedure or process can be derived from the functions of its parts. Then, the derived (or abstracted) function is compared with the intended function (specification) to determine if the procedure is correct with respect to its specification. The goal of this section is to introduce techniques that support the use of the functional model in verifying Cleanroom designs that use objects. Warning: some of these techniques are not pretty! However, we will show that they are necessary in order to support true program verification. The most important part of correctness evaluation, and the hardest to get right, is to choose which subspecifications to document. If too many are documented, then a productivity price is paid in maintaining subspecifications that restate the obvious effect of the code. If there are too few, then the review team must derive intermediate subspecifications in real time in order to do an adequate job of review. If those subspecifications are not recorded, then subsequent re-reviews must again re-derive the functions. One common standard among Cleanroom teams has been to recommend that the function of every procedure-call be documented. We might call this activity "flattening" since it has the effect of producing a module that has on off-page references to other functions. Such a module canideally be verified in a vacuum, without knowing either the context in which it is invoked, or the subprocedures that it relies on. Flattening presents more of a challenge in the object-oriented world but it is even more important there. We will look at techniques for flattening, beginning with simple procedures and ending up with subclasses and polymorphism.

4.5. Application Frameworks : Special Issues

Frameworks are class libraries that simplify the construction of specific kinds of

teams, and gradually phasing in history (if necessary) after the team has some experience. Whatever has motivated us to do so, we may choose to organize specification around a set of abstract attributes other than stimulus history. The form of the black box description reflects this choice, as in Figure 5.

Stimulus	Condition	Response	Model Update
check out a car to renter	pool not empty and renter is qualified	OK, car number	pool : = pool less car selected
check out a car to renter	pool empty and renter is qualified	Sorry, no cars	no change
check out a car to renter	pool not empty and renter not qualified	Sorry, not qualified	no change

Figure 5 : Car- Rental System Specification (Fragment 2)

Now that we have apportioned the abstract conceptual model of the entire system among conceptual objects (and, potentially, classes, though space does not permit us to expose these details) at the specification level, we can consider the methods we are applying to those objects (e.g., pool. remove - car; pool. is - empty) and the effect of those methods on the abstract attributes of the object. Again, this is not the same as state box design, since the object "pool" is not necessarily identified with any state data - we could choose to implement the concept of a "pool" object as several discrete objects, or further apportion its functions throughout a more comprehensive state data object.

Applying transction closure to the richer models is the same as applying it to the history model. We must ensure that every attribute is equipped to process any method we apply to it. So, when we wrote pool. is - empty, we had to ask whether the attribute pool has been initialized properly to give an appropriate response.

4.3. Finishing the OO/Black Box Specification

The process of completing the OO/black box specification alternates object analysis with black box defintion. The purpose of both is to arrive at a complete description of the externally-visible of the system that can be easily validated against the requirements and that gives appropriate guidance to designers. Space does not permit a lengthy discussion of techniques to be used in validating the specification-

Stimulus	Condition	Response
Check out a car to renter	cars are available and renter is qualified	OK, car number
Check out a car to renter	cars not available and renter is qualified	Sorry, no cars
Check out a car to renter	cars available and renter not qualified	Sorry, not qualified

Figure 4 : Car- Rental System Specification (Fragment 1)

Our primitive statements about "cars available" are understood to be predicates on stimulus history that we will want to understand and documents as such. At some point, we will have accumulated enough knowledge to decide that the concept of a "pool" of cars is important, and we can write a function on stimulus history that returns a set of cars - the current pool. We could then rewrite the specification in Figure 4 using "is - in pool (history)" rather than "cars are available." As we start asking more questions about the car-rental system, we would soon discover that the cars themselves abstract attributes. For example, a renter might ask for a particular kind of car, and we would need to extract a pool subset of cars having that attribute. This kind of analysis might lead us to a hierarchy of car classes, where the base class has certain attributes (e.g. Current Fuel Level) and the derived classes (e.g. mid-size) have additional attributes. We can continue to describe car classes and their attributes, all within the stimulus history framework.

One thing that separates this model from a true OO viewpoint is there are no "write" methods or operations on the abstract attributes. There's a very good reason for this - the only "real" entity in the conceptual model is stimulus history. So, there is no way to change the "pool of cars" directly - the history is changed by a stimulus from outside the system (e.g. check out) and then specification functions on history enable a view of the pool of cars. As we notice, this feature of the history model has its advantages and disadvatages.

4.2. OO/Black Box Specitication Beginning with Objects and Classes

When the specification activity starts with a richer conceptual model already in place, it may make more sense to begin with objects and add the black box framework to it. We may also decide, after some initial effort, that the history-based specification is growing to be too cumbersome, especially if a conceptual model has already been established. This is an occurrence that, in the author's experience, happens frequently with Cleanroom teams that are unfamiliar with black box analysis, pointing us toward recommending the more generous abstract models to new

and information hiding are included in the broader idea of abstraction.

- Messages are the primary, if not sole, information-transport mechanism among objects. Within the broader category of messages, we include their various implementations, e.g., call/return and messages queuing.
- Inheritance defines a particular relationship among objects or classes, namely the ISA relationship. Distinction among the different forms of inheritance is not essential to the understanding of the ideas in this paper.
- Polymorphism means the ability to use the same expression to denote different operations. It includes both operator overloading and genericity.

Any number of OO methodologies [11, 12, 13] have evolved to turn these abstract concepts into workable software design paradigms. This paper favors no particular methodology or tool.

4. INTEGRATING OBJECT-ORIENTED ANALYSIS AND BLACK BOX SPECIFICATIONS

Object-oriented analysis can be alternated with black box techniques to arrive at a specification that combines the best of both approaches. The black box techniques focus on describing the externally-visible behavior, while OOA focuses on organizing abstract attributes for ease of understanding that description. Where to begin depends primarily on how well-formed a set of conceptual objects already exists in the initial requirements or in the physical system being modeled. How specification evolves will be considered, starting from both history and objects.

4.1 OO/Black Box Specification Beginning with History

Most box-structures examples begin with very general understanding of the conceptual state and a list of stimuli that must be accepted and processed. This situation might present itself in a transaction-processing enviroment, such as the car-rental service example described by Hevner [25]. That system was expected to respond to requests such as "check out a car," "return a car," and so forth. At such an early level of discourse, not much is known about conceptual objects except for the basic object "car." As we progress with a pure stimulus-history analysis, we see that the responses to "check out a car" depend on system attributes like "cars available." An initial specification sketch might have a fragment such as that shown in Figure 4.

This form of correctness is called "sufficient correctness" since we have not required that the domain of the actual specification be exactly the same as that of the intended specification.

2.3.5. Design Summary

Although the emphasis of verification appears to be on design, a design is nothing more than a structure linking carefully chosen and documented specifications. Thus, the central skill of Cleanroom development is the ability to write a good specification for an object of any size. One of the most challenging parts of Cleanroom development is maintaining these specifications. Almost any code change will necessitate at least a small amount of specification change; a code change that affects the user can require specification changes at many levels, as higher and higher levels of specification are changed in order to maintain the chain of verification. However, maintaining these specifications has many benefits, including: visibility of possible ripple effeccts, re-examination and evaluation of high-level design decisions, training members who have just joined the team, and improving everyone's proficiency at review. Indeed, verification is easy when specifications are good. A learned Cleanroom skill is writing specifications that give only enough detail to support verification, yet hide enough that changes do not propagate unnecessarily upwards through the system.

Each step of box structures design introduces new black boxes. The process is complete when all data and processes are expressed in terms of black boxes as shown in figure 2, that are already implemented, either by the programming language, by reused code, or by environment objects. Many languages encourage reuse by permitting processes and data to be made generic for instantiation in different parts of the system. In box structures these are called "common services". Procedures, with or without parameters, are examples of common services, as are shared data objects. Classes (in object-orientation) are a higher-order form of common services, where a number of objects have the same specification.

3. OBJECT ORIENTED SOFTWARE ENGINEERING

Object oriented software engineering is defined as a software engineering process that incorporates several concepts [10].

- Objects are the basic units of construction, whether in concept or in actual programming. Objects are organized into classes that have common attributes and/or operations.
- Abstraction hides implementation details of an object (or class) from other objects in the system. The related, though not identical, notions of encapsulation

the black box, whenever the black box's behavior is defined. While there is a great deal of mathematics behind the rigorous defintion of correctness, its presentation is beyond the scope of this paper.

2.3.3. Process Design : The Clear Box

The third view of a system, object, or part is the clear box view. In this view, procedural details of ordering and local data are exposed. As with the state box, this is a partial exposure, as new specifications can be introduced. The common units of structured programming sequences, altermations (if then and if thenelse), and iterations (whiledo, dountil) are the building blocks of the clear box. we did not have to begin with a data-abstracting black box in order to see a clear box -- any process abstraction will first be designed as a clear box.

2.3.4. Clear Box Correctness Verification

Because the step from process specification (black box) to clear box is a one-to-many design step, it too requires verification of corrrectness as well as design quality assessment [24]. Figure 3 shows the general theory of clear box verification. Given an intended process specification and a proposed clear box design, an abstraction operation is done to yield the actual behavior specification of the design. The two specifications are compared and, if the actual specification sufficiently meets the intent, the design is correct. Again the mathematical reasoning to support this claim can be found several sources. The fundamental issues of correctness are:

- Is the domain of the actual specification at least the domain of the intended specification? That is, is the actual function defined whenever the intended function is defined?
- For every element of the domain of the intended specification, does the actual specification produce the same result as the intended specification.

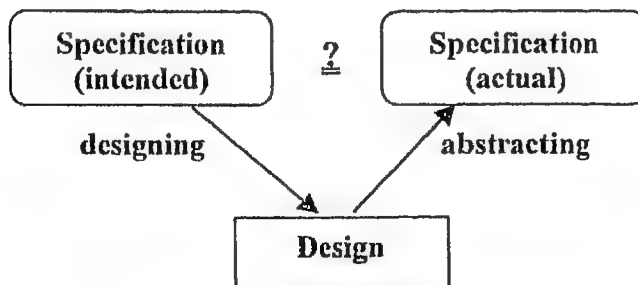


Figure 3 : Schematic of Clear Box Verifcation

2.2.3 Data Specifications

In the early work on specifications [8], all of the data objects were "pure mathematical" integer data objects. However, we know that most programs use more complicated data objects. One of the essential characteristics of any data object is the mental picture we have of what is "inside" it. We can use this mental picture, or abstract model, to understand and manipulate the object. This abstract model permits us to discuss the value of the object without knowing its internals. An abstract model defines a new data object in terms of more primitive or better-understood concepts. The black box of a data-retaining object, to which we will refer as a data specification, consists of an abstract data model and a collection of process specifications that can be used to manipulate it. These process abstractions are sometimes called services, transitions, or methods. The definition of an abstract data model and all of the operations on it constitutes a data type defintion, from which we can instantiate data objects. The standard box structures literature takes this a step farther and suggests a view of the abstract model that consists of only one entity the history of all inputs received by the object. This entity, called stimulus history, has its practical advantages and disadvantages [23].

2.3 Design : A Hierarchy of specifications

As the specification nears completion, the design activity begins. An object being specified as a process abstraction is designed first as a clear box; a data abstraction is designed first as a box. Each design reveals new black boxes that may require further design. Because the black box specification ideas can be applied to both high-level (large) objects and low-level (small) objects, this hierarchy of specifications is self-similar throughout the development process as shown in figure 2.

2.3.1 Data Disign : The State Box

The state box expresses the abstract model of a data-encapsulating system in a more concrete form, in terms of objects that are either simpler to implement than the entire system or that may already exist. The state box has two conceptual parts. The state data is a collection of data objects and the machine is a process specification that has access to the black box inputs as well as to the state data. There are two important notes to be made. First, the black box and state box are different views of the same system. The black box and state box have the same inputs and outputs. The state box view exposes some concrete data aspects, and exposes no operations-ordering or other concrete process artifacts. The second note of importance is that the step from black box to state box is a one-to many design step.

2.3.2 State Box Correctness Verification

A State box is correct with respect to a black box if it has the same behavior as

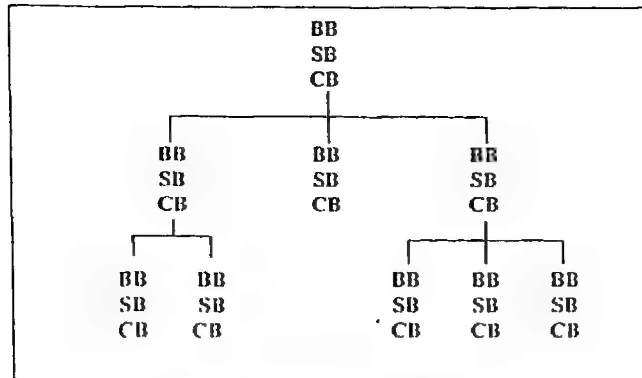


Figure 2 : Box Structure Hierarchy

2.2.2 Process Specifications

A process is a definition for how states are transformed by the computer. In the functional model, a process is defined by a mapping from "before" states of the system into "after" states. This mapping is called a process abstraction or process specification. There are several aspects of these specifications that are important to Cleanroom, independent of notation, conceptually, at least:

- the evaluation of all the "before" values is done before the process begins execution.
- assignment of "after" values is done all at - once with no intermediate steps.
- only the "before" values can be used to compute the output and there are no undocumentes "global" data objects.
- there are no undocumented side effects, any data object whose value is changed must be described.
- a response must be produced for one input before the next input can be accepted.

If these rules are followed, the process specification hides the order of operations and all procedurality, as well as local data of the algorithm and is completely independent of its usage context. The specification of a complex such as a batch compiler is necessarily long and detailed, but can done using this same idea. The inputs would be the contents of source files, options, and command-line parameters. Outputs would include the contents of object files, listings, and screen messages. Environmental structures such as memory and other resources can be treated as both input and output. Graphics are often used in teaching box structures and in informal discussions among developers. Their usage is somewhat more rare in large-scale development efforts.

2.2 Box Structures : A Formal Model for Cleanroom

As Cleanroom evolved, so did the formal models used in Cleanroom development. The earliest Cleanroom projects used the functional model [17, 18]. However, the functional model deals with the correctness of procedures; it does not have specific techniques for dealing with data. Cleanroom teams used the method of state machines [19, 20] where data verification was important. An additional element was added, and an overall stucture proposed, by Mills [21, 22], with the result called "box structures". Since the resulting framework incorporates both functional verification and state machincs as special cases, it is not incorrect to say that box structures has always been the foundation for Cleanroom.

The box structures method as shown in figure 1, provides the software developer with three different views of a software system, object, or part. The black box view hides all of the object's implementation details, including any implementation or processing. The state box view partially exposes the data implementation while continuing to hide procedurality. The clear box view partially exposes procedurality. Each of these last two views may include references to new black boxes, defining a usage hierarchy as shown in Figure 2. The box structure usage hierarchy results from stepwise decomposition of black boxes until no new black boxes remain. The usage hierarchy is orthogonal to the inheritance hierarchy of object-orientation.

2.2.1 Specification : The Black Box

Two kinds of black box specifications are separately considered: process specifications and data specifcations. A process specification is the black box of a software entity that does not retain data, it does not behave differently at different times depending on prior usage. Process specifications would be used to specify procedures, fragments of inline code, or batch applications. A data specification, on the other hand, is used to represent an object that does encapsulate data, such as a file, database, or object (in the object-oriented sense).

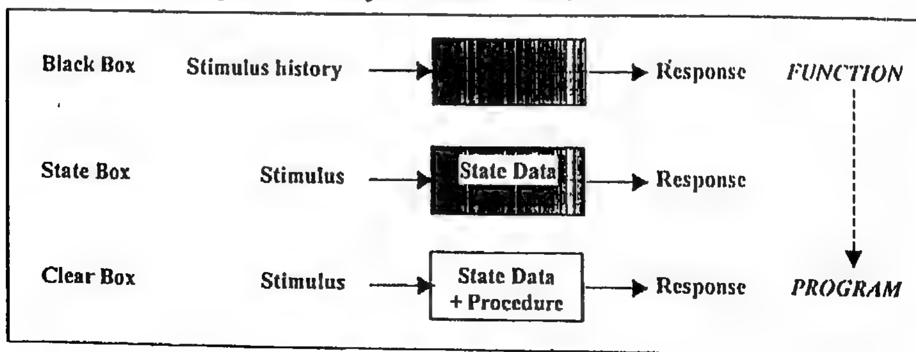


Figure 1 : Box Stucture Method

dressed by a parent class. It is the class concept that is the driving force behind the composition of systems from reusable components.

Booch describes the underlying modèls (meta-model) upon which all object-oriented methods are based [14]. He identifies the major elements in this meta model as the software techniques of abstraction, encapsulation, modularity, and hierarchy. The minor elements of the meta-model are typing, concurrency and persistence [14]. These elements form the conceptual framework for the development of classes that represent the behaviors and properties of an object type and their integration into a system. Also identifies three categories of methods that are used to support system analysis and design: top-down structured design, data-driven design, and object-oriented design [14]. Top-down structured design is algorithmic decomposition. Data-driven design derives the structure of software systems by mapping system inputs to system outputs.

One of the most important characteristics of object-orientation is its focus on the behaviors of the system we are to develop, and the behavior of its objects through this descrption of object behaviors, that we define the stimulus sequences, and responses involved in a system communicating with external objects and among communicating objects within the system.

The discipline called "object-oriented analysis" has recognized the importance of understanding the behavior of the software systems and objects of which they are to be composed, Booch, Shlaer, and Jacobson [14, 15, 16] all identify the need for performing analysis of and developing models to describe the behavior of a system. It is interesting to note that one of the most important aspects of Cleanroom is the development of an implementation-and state-free behavioral specification for a proposed system and each of its objects (black boxes).

2. THE CLEANROOM SOFTWARE ENGINEERING

2.1 Fundamental Principles

Although some of the specific practices have changed over time, the fundamental principles of Cleanroom have remained very much the same [17]. First; programming teams can and should strive to produce systems that are nearly error-free upon entry to testing. Second; the purpose of testing is to certify the reliability of the developed software product, and not to "test quality in". Third; incremental team-based management practices allow in-process feedback for continuous improvement, and limit the scope of human fallibility.

The demonstrated success of the large-scale users of clean rooms, especially the makers of integrated circuits, prompted Dr. Harlan Mills of IBM's Fedral Systems Division (FSD) to suggest that the same principles' defect prevention during design, separation of design and test, and objective testing criteria, be applied to software development. Although the Cleanroom name, and some of its practices, were first published in 1981 [5], it was not described in major journals until 1986 [6] and 1987 [7]. Together, these papers describe Cleanroom practices at that time.

Beginning in 1987, a number of other organizations, in addition to IBM, began to apply Cleanroom techniques [8, 9]. Cleanroom have been evolved to keep up with the changing world of software. Users of Cleanroom have adapted it to coexist with a variety of tools and techniques. Several consulting organizations have been formed to help teams use Cleanrooms, and Ph.D. Dissertations have addressed special issues with the methodology. The keynote phrase for the large-scale adoption of Cleanroom hinges on the idea of an "expectation of quality". Cleanroom represents a fundamental shift away from the notion that errors are likely and frequent.

Object-oriented development is about analyzing and implementing systems that comprise collaborating objects, where each object encapsulate the data methods necessary to satisfy its processing requests. Object-orientation emphasizes the specification of the external interfaces of objects, and requires the practice of information hiding and the encapsulation of functions and data that perform the work of the object [10]. Objects provide convenient concept in which to think about the composition of systems. System composition through object requires thinking about the architecture for a system in terms of assembling systems through the use of both large and fine graines components. Viewing systems as a composition of collaboration objects, also supports the idea of developing not just reusable assets, but domain- specific solution architectures for classes of problems, where architectures, as well as objects, become units of system development and integration.

In pracitce, even though an enormous amount of software has been developed, only a portion of it is recoverable from "mining and defining" efforts. As software is developed in the future, and good software engineering techniques are employed to define and develop robust software objects, we may one day build up a sufficient quantity of software objects such that systems can be composed from reusable components, and software development may become more of an integration acitvity than one of development. One of the most important contributions of OO methods is the concept of developing reusable classes, which through inheritance, may be specialized [11, 12, 13]. This speciatizing of generalized classes and their methods permits some methods to be inherited without modification, and allows others to be specialized as necessary to satisfy unique processing requirements not ad-

CHAPTER 4

IMPROVING SOFTWARE RELIABILITY THROUGH INTEGRATED CLEANROOM AND OBJECT ORIENTATION DEVELOPMENT

Dr. Alaa Mohamed Fahmy

ABSTRACT

Cleanroom software engineering is an approach to software development that improves quality and reduces cost. The approach takes its name from the clean rooms used in chip manufacturing, where statictical quality control techniques emphasize defect prevention over defect removal. Cleanroom focuses on software reliability, but does not prescribe design techiques. Object-Oriented (OO) development approaches offer techniques for developing designs and architectures that are robust, reusable, and maintainable. The combination of these two approaches leading to an integrated Cleanroom/OO development process stronger than either individual process. This combination offers reliability through correctness verification based on the functional mode of Cleanroom ads well as design guidance characterizing OO development. This paper shows how to integrate object-oriented analysis with Cleanroom black box specification, suggests documentation needed to support verification of OO designs. A guidance in making tradeoffs during the integration process is also presented.

1. INTRODUCTION

Cleanroom software engineering is a set of principles and practices for software management, specification, design and testing that have been proven effective in improving software development quality while at the same time improving productivity and reducing cost [1, 2, 3,]. The name Caleanroom comes from the clean rooms used in high-precision manufacturing [4]. There many similarities between Cleanroom software engineering and the hardware clean rooms after which it is named.

REFERENCES

1) W. W. Agresti, New Paradigms for Software Development, IEEE Computer Society EH- 0245-1, Washington, DC (1986).

2) S. Skimore, B. Wore, Introducing Systems Analysis, NCC publications, Manchester, England (1988).

3) R. Wieringa, Three Roles of Conceptual Models in Information System Design and Use. In proceedings of the confernce on Information Systems Concepts, E. D. Falkenberg and P. Lindgreen Namur, (October 1989).

4) T. W. Olle, el Information System Methodologies- A Framework of Understanding. Addison - Wesley, (1988).

5) V. Stuart, Equivalence Conditions for Information Systems Representation, Report 87-09A, Department of Computer Science, University of Leiden, Netherlands (1987).

6) S. Brinkkemper, The Essence and Support of Modelling Transparency, 3rd International Conference on CASE, Imperial College, London, UK, (July 1989).

7) E. D. Falkenberg, Information Modelling - Subjective Forever, Proceedings Database. Dag, K. M. van Hee and T. M. A Bemelamns (Eds.) Eindhoren (Dec. 1989).

8) R. J. Mayer, Information Integration For Concurrent Engineering (IICE) IDEFA Object- Oriented Design Method Report, Airforce Systems Command. W.P Airforcebase, OHIO. (1992).

9) R. J. Mayer, IDEF3 Technical Report. WPAFB, OH, USA (1991).

10) C. Gane, T. Sarson, Structured Systems Analysis: Tools and Techniques, Prentic Hall, E. C. (1979).

11) J. Martin, Information Engineering, Vol- 1,2,3, Savant Research Studies, Lanshire. (1988).

9. FUTURE WORK AND EXTENSION

This paper may be considered as an introduction to a series of research work concerning the formalization of information systems. Although, there exist a variety of methods, techniques and tools which can be used during the system development life cycle, their wide range of rules, and use may now cause problems than they may solve. Mathematical modeelling needs precise definitions of every and each aspect of information system developing process.

In this context, further work is currently done to include meta modelling, mathematical data models, models of context diagrams, Data Flow Diagrams, Tasks, Processes and Activities.

This clarifies the use of models during the development of information systems and their study with the help of meta-modelling. [7]. This will lead to the following definition as the very first step to use the set theory as the basis of a "mathematics" of the analysis phase.

Definition (7) :

"A Universe of Discourse (UoD) is a system of concrete entities which were, are, or will be relevant to a given objective. An information system is a manual, partially automated, or fully automated system of symbolic entities, representing facts about concrete entities, that are recorded because of their relevance with respect to a given objective, and that can be updated, retrieved, and from which other facts, can be derived". Figure 4 clarifirs the above definition.

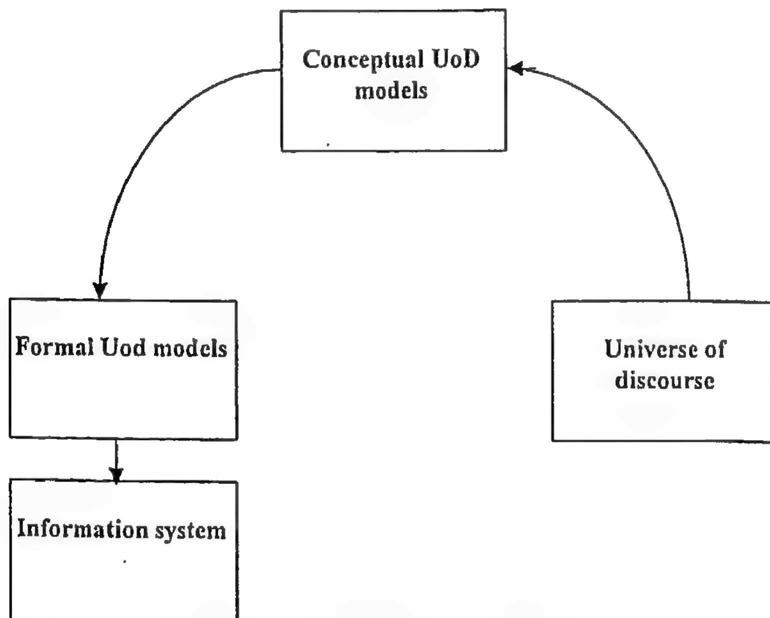


Figure 4 : Information systems modelling

8. FORMAL VISUALS

Many diagrammatic techniques are used for representation in the information modelling process. An overview of the most popular ones can be found in [11]. A modelling technique is a visual formalization if it is based on mathematical theory with a corresponding graphical notation, i.e. the existing techniques can be extended with a rigorous definition of their semantics which formalizes the visual techniques.

7. MODELS OF INFORMATION SYSTEMS

It is known [3] that there are 3 classes of systems:

- *Conceptual systems* : Systems of which entities are concepts or things in the minds of people (e.g. Set theory).
- *Concrete systems* : Systems of which entities are concrete objects (e. g. hardware systems).
- Symbolic systems : Systems of which entities are uninterpreted syntactical symbols (e. g. C - language).

From definition (1), models are systems stand in a particular relation with another system, so we may have nine possible cases, only four of them will be considered as shown in figure 3. It shows the nine possible transformations among the 3 types of systems. Each is a model type. We will consider: conceptualization, mapping, representations, and translation.

- *Conceptualization model* : Such a model may result after the observation of a concrete system. It is the heart of the analysis phase.
- *Mapping model* : It is the conceptual model of a conceptual system (e. g. transfer from DFD [10] to TSAC technique [11].
- *Representation model* : It is the symbolic model of a symbolic system (e. g. the activity model of the DFD).
- *Translation model* : It is the symbolic model of a symbolic system (e. g. the transfer of data flow activity to the ISAC activity technique [11]).

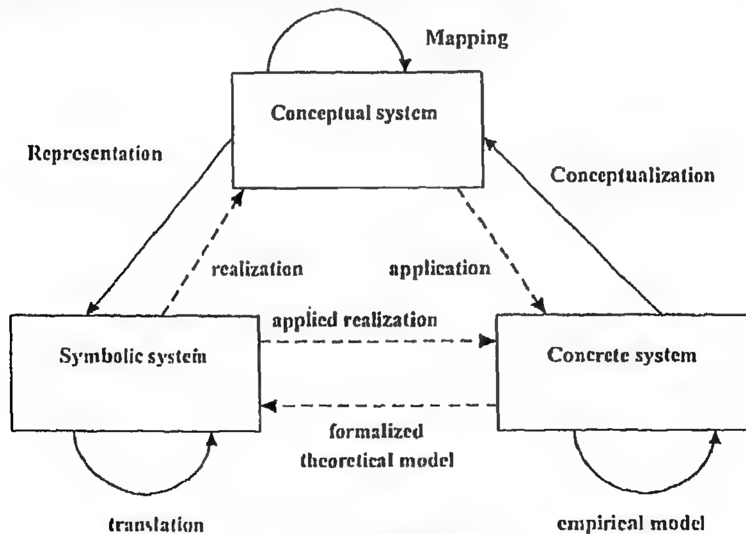


Figure 3 : The model triangle

6. A VIEW TO METHODS, TECHNIQUES, AND TOOLS

Definition (2) :

"The methodology of information systems development is the systematic description, explanation and evaluation of all aspects of methodical information systems development" [8].

Even so, methodological schools differ: SWE, DBMS, MIS, and infological approach, It is a differentiation which can be considered as one of the causes of software crisis.

Definition (3) :

"A method is an approach, based on a certain way of thinking, to carry out an information system development process, consisting of directions and rules structured according to a systematic ordering of development activities and corresponding development process" [8].

Examples are: ISAC, IEM, and SDM [9].

Definition [4] :

"A technique provides the description of the manner in which, and the notation with which a part of the development must take place. This incorporates the practical steps to follow when carrying out a development" [8].

Examples are : process decomposition, affinity analysis, change analysis, and group interviewing [9].

Definition (5) :

"A notation is a system of symbols with a corresponding set rules, which determine the correct application of the symbols. A notation is used to denote the results of a technique" [8].

Examples are : decomposition diagram, association matrix, and NIAM [9].

Definition (6) :

"A tool is a possibly automated mean to carry out a part of the development process. A tools may support a notation, a technique, or even a method" [8].

Examples are : Designer 2000, Developer 2000 of Oracle, Power Builder of Sybase, and Newera of Informix [9].

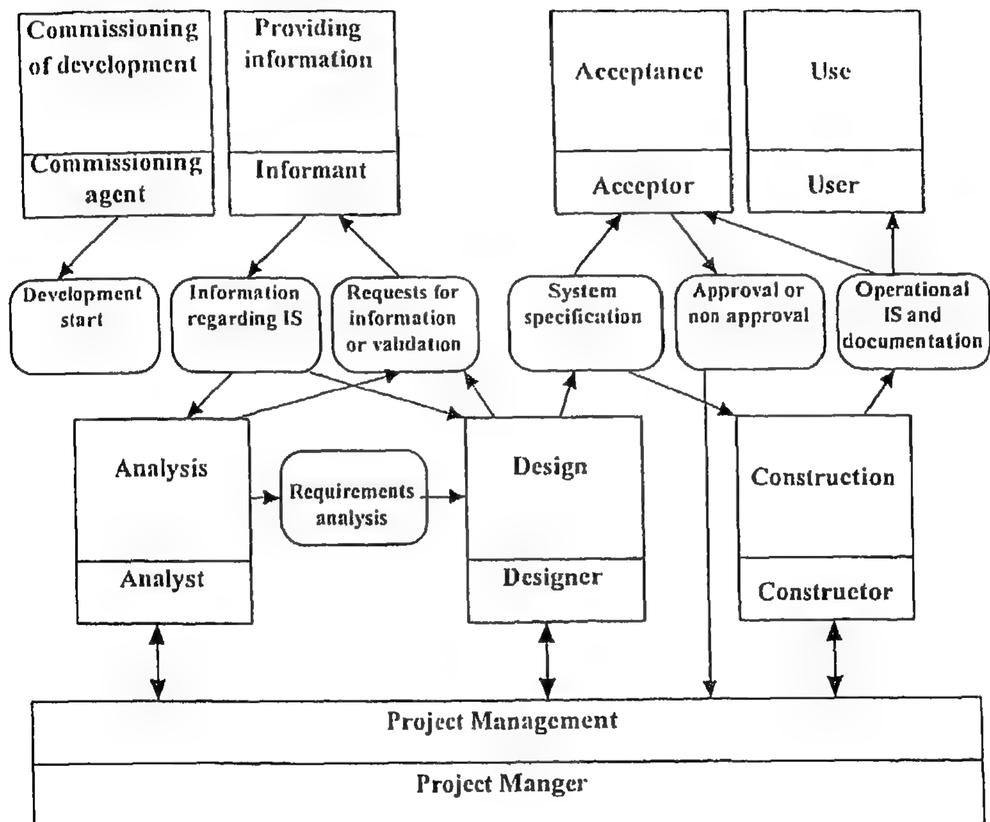


Figure 2: Process structure and cast of a development project

- What is the best way to construct a model, when a specific technique is applied?
- What is the formal basis of the modelling technique?
- How is one particular model related to other models of an information system?

then we would realize a set of goals to get rid of the problems mentioned in Section 3. The starting point for making the modelling techniques more explicit is the claim that the modelling techniques more explicit is the claim that the modelling process can be more formatized than it is now.[6].

5. THE HUMAN ROLES IN DEVELOPING A MODEL FOR INFORMATION SYSTEM DEVELOPMENT

Figure 2 shows a view to the roles of people take parts in modelling information systems. The lower part of the figure overviews the roles of developers while the upper parts describes the activities of the environment. The following definitions are given to each participant [4, 5]:

a) *Commissioning Agent:* A person who has the responsibility of giving the task to devlop a system and the assignment of other roles in the environment.

b) *Informant :* A person who is specifying information about the organizational and the hardware requirements and about the data to be stored in the information system.

c) *Acceptor :* A person who is responsible of the approval of the complete intermediate requirement specifications and the final operational system.

d) *User :* A person who plans and manages all development activities and the related resources.

e) *Analyst :* A person who performs the analysis stage of the project in which the system scope is defined and the current system in described (the modeller).

f) *Designer :* A person who develops the specifications of the new system (the modeller).

g) *constructor :* A person who constructs and documents the information system on the basis of the requirements' specification.

Diagrammatic formula modelling is shown in Section 8, while Section 9 proposes future research.

2. MODELLING

Developers make use of models more extensive in the mapping phase than in the conceptualization phase. Variants of most model types are defined in relations to the various development stages.

Definition (1) :

"A system A is used as a model to obtain knowledge about system B, where the system A is neither directly nor indirectly interacting with the system B".[3]

Models, through the history of information system development, are used widely for processes, data, databases, and interfaces. Tools such as diagrams, schemes, tables, and programs are used to create a communication language among developers across the life cycles of systems.[6]

3. THE PROBLEM DEFINITION

The current modelling processes in information systems are suffering too many problems:

- Different models are constructed for the same system at hand at the diferent development stages.
- The modelling process takes too long.
- The models are not consistent with the systems they are supposed to model.
- The gathering of information's specifications is hard.
- The modelling techniques are vague and unformalized.
- Experienced modellers are scarce and expensive.
- Automatic modelling tools are not exist especially in the early conceptualization phase.

4. THE SOLUTION APPROACH

Abstract modelling and standardization is an approach to solve the problem of informality during conceptualization i.e. make the modelling process of the techniques applied during information systems development more explicit. If we can answer the following questions:

there are general problem solving techniques: conceptulization which is the creation of a complete and formal system specification starting from the informal rough specification, and mapping which is the transformation of the specifications on a conceptual level (i.e. without presentation and implementation details) to the machine executable specification.

Figure 1 shows that development can then be depicted in a pathway in the 2 dimensional areas. The ideal path is to go sharply from the informal point to the complete formal point and then sharply vertical to the machine level point. Any other path especially the opposite of the above will cause a problem. This set of problems was defined as the software crsis i.e. when large part of the conceptualization process is performed on the mapping level.

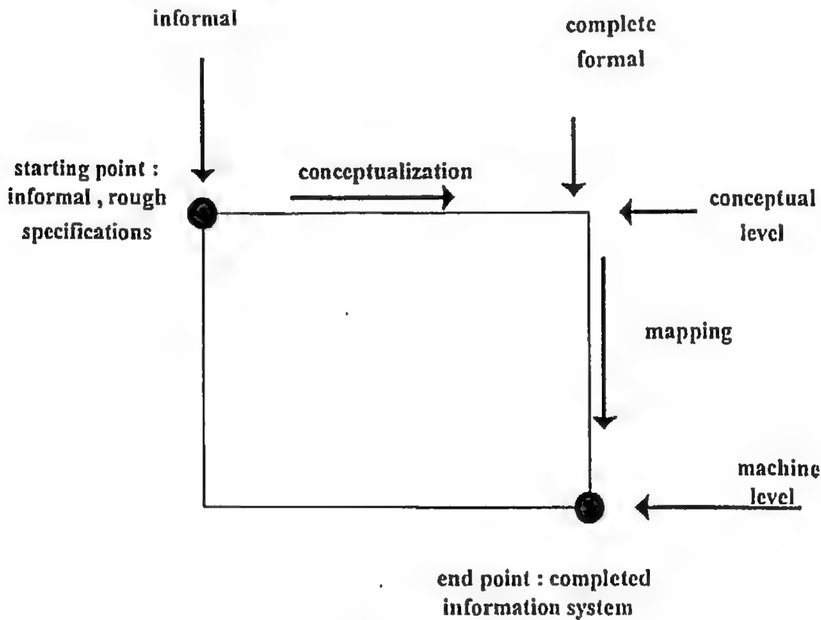


Figure 1: conceptualization and mapping

In this paper, we give a defintion of a model from the point of view of information systems in Section 2, while the problem we try to tackle is defined in Section 3, and proposed approach in Section 4. In order to automate the analysis phase the manual procedures and their performer's roles should be precisely defined. The task of Section 6 views and definitions of terms like: method, technique, and tool are shown. Section 7 describes the relations between different types of models and systems and where a model for an IS will be presented.

CHAPTER 3

An Abstract Modelling for Information Systems: A first step to automate the analysis phase

By

Dr. M. B. Senousy

ABSTRACT

Software is becoming more and more complex to be developed. New technological advancement and increasing user requirements demand large numbers of designers and programmers, good cooperation, and experienced project management for effective and efficient software development.

Despite the resources employed in the software development especially in the design phase (object orientation, highly specialized design tools: Newera with Informix, Power Builder with Sybase, and Designer 2000 with Oracle), the malfunctioning of automated systems and projects running out of budget re regular news. The software crisis is a frequent subject of the new media all over the world.

In order to overcome this case in the future, improved methods and techniques for software development will solve the crisis by increasing the quality of the analysis phase and hence of the developed software systems.

This paper is aiming to launch, formulate, and give precise definitions of micro steps and micro entities during the analysis phase to reach a frame of a model for an information system. The paper shows the conceptualization and the mapping prosesses, define the "modelling" from IS point of view. This research includes the abstraction of the human rules in information systems development. Also, a refinement view of methods, techniques and tools are shown. Finally a model of an IS is given.

1. INTRODUCTION

More than 80% of computerized applications are MISs in their various forms.[1] Methodical information system development is mostly performed in stage such as: scope definition, analysis, design, construction, and use[2]. During development,

PART TWO

SOFTWARE DEVELOPMENT ENGINEERING: MODELS, TOOLS AND TECHNIQUES

CHAPTER 2

The National Bibliographic Control of Computer programs in Egypt*

Dr. Mostafa Husam El-Din

ABSTRACT

Three types of bibliographic tools are required for national bibliographic control of computer programs.

The first tool comprises bibliographic records that identify every computer program produced in the country, describe its content, and give access to facilitate acquiring, and dealing with them.

The second tool gives, in addition to the bibliographic records detailed information about the special characterstics of every program as given by the producer and/ or distributor, with extracts from the most prominent reviews on them.

The third tool covers all what have been written on every program. So, it helps in identifying every article, paper, and even book written on every program.

No one of these bibliographic tools is available in Egypt, except some incomplete data or advertisements in newspapers and computer magazines.

So, this paper presents a vision of the prerequisites of national bibliogaphic control of computer programs in Egypt: the priorities, the responsible and authorized agency, the standard bibliographic data, sources of collecting data, access or searching points, availability of the data in terms of forms and formats, and continuity and regulation of dissemination these data.

* The Original Text is written in Arabic Language

ton, DC., 1993.

[17] Heather Hudson, "Economic and Social Benefits of Rural Telecommunications: A Report to the Workd Bank, World Bank, Washington, DC., March 1995.

[18] Satyan Pitroda; "Development Democracy, and the Village Telephone", Harvard Business Review, Nor/ Dec, 1993.

[19] Intelecon Research & Consulancy Ltd; "Option for Rural Telecommunications Development, World Bank Report, April 1995.

[20] Simon Forge; "Consequences of Current Telecommunications Trends on the Competitveness of Developing Countries", World Bank Discussion Paper, January 1995.

[21] Volunteers in Technical Assistance (VITA); "Community Information Centers", Arlington, VA, January, 1995.

[22] World Bank; "The East Asian Miracle: Economic Growth and Public", Oxford University Press, New York, 1993.

[23] Gary Stix; "The Speed of Write", Scientific American, December 1994.

[24] Marcia Parish & William Junk; "Building the Information Systems Development Capability of Public Institutions in Sub-Saharan Africa", In Linkage and Developing Countries: Information Processing 94, Vol. 3, 1994.

[25] Marcia Zielenziger; "Dream Team?" For Economic Review, May 4, 1995.

[26] Antoun Mousse & Robert Schware; "Information in Africa: Lessons from World Bank Experience", World Development, Vol. 20, No. 12, 1992.

REFERENCES :

[1] Nagy Hanna & Sandor Boyson; "Information Technology in World Bank Lending: Increasing the Development Impact". World Bank Discussion Papers # 206, World Bank, Washington DC., 1993.

[2] Edward Yourdon; "Developing Software Overseas", Byte, June 1994.

[3] William Montalbano; "Culture: Egypt's Sleeping Beauty Revived After 3.200 years; Restoring the Tomb of Queen Nefertari Raises Profound Issues of Art and History", Los Angeles Times, 18 October, 1994.

[4] Carlos A. Prime Braga; "The Impact of the Internationalization of Services on Developing Countries", Finance & Development, issue of March 1996.

[5] Peter Druker; "Trade Lessons from the World Economy", Foreign Affairs, vol 73, No. 1, Jan/Feb 1994.

[6] Peter Draker; "The Age of Social Transformation", The Atlantic Monthly, November 1994.

[7] Cirano Iochpe; "Software Incubators in Latin America: Experiences and Prespectives", UNIDO Microelectronics Monitor, Vol. 1, No. 3, 1994.

[8] Eduardo Talero; "A Demand-Driven Approach to National Informatics Policy", IFIP 13th World Computer Congress, Vol. 3, Elsevier Science, 1994.

[9] Peter Haynes; "Survey of Telecommunications", The Economist, 23 October, 1993.

[10] Peter Haynes; "Survey of Computer Industry", The Economist, 17 Sep, 1994.

[11] Adrian Wooldridge; "Survey of Education", The Economist, 21 Nov., 1992.

[12] Hilary Perraton; "A Review of Distance Education", EDI Development Policy Case Series, Analytical Case Studies, No. 9, World Bank, Washington DC., 1992.

[13] Globe Program, Internet: http//globe. fsl. noaa. gov

[14] Robert Schware; "Information Technology and Development", World Bank, January 1995.

[15] Nagy Hanna; "The Information Technology Revolution and Economic Development", World Bank, Washington DC., 1991.

[16] Nagy Hanna; "Exploiting Information Techology for Development- A Case Study of India", World Bank Discussion Papers # 246, World Bank Washing-

An extreme problem of managing change affects individuals, firms, professions, and countries. Even in the best of circumstances, the transition will put some industries and countries at a severe disavantage. This transformation of employment reinforces the urgency and strategic importance of education and the need for lifelong learning at all levels.

Both theory and evidence suggest that in the long run new technology should create more jobs than it destroys. But the long run can take a long time. Unfortunately there may be prolonged lags between job losses and the creation of new jobs. And the new jobs may anyway be inappropriate for the displaced workers. Governments can help by making workers more adaptable through improvements in education and training and by removing obstacles to free markets in labour and in goods and services.

8 . CONCLUSION

The Software industry of Egypt may have a chance of becoming the Second Garments Industry, both in terms of the social change that it can bring and also the economic boost that it can give to our economy. The industry has the highest possible ROI. The growth of training centers also points to the fact that there is a healthy demand from the people as well as the industry. We should and we can sell our Software worldwide, but there is a long way ahead.

We need to educate more people and support lifelong learning. Make governments more efficient, accountable, and transparent. Increase effectiveness of economic reforms. Monitor and protect the environment. Reduce information and income inequalities. Overcome natural disadvantages. Diversify into information-intensive industries. Promote small and medium enterprises. Participate in global trade and competition.

Also, an information-friendly environment must be created. An information infrastructure must be deployed. New demands for education and training must be met. Sectoral information networks must be developed. Consortia for advice, investment, and research must be created.

Constraints and hazards of information society must be recognized. Social and technical complexity. Waste and misallocation of scarce resources. Cultural preservation and social disruption. Social and employment inequities.

Bank projects show mixed experience with information in Africa. Government agencies are the dominant users of information technology (information technology) in Africa. As in the industrial countries, the returns of the use of information technology are mixed. Analysis of 76 World Bank projects in Africa including Egypt shows that government and aid agencies alike need to make more intensive efforts in the design of projects with information technology components. Specific examples of information technology use in Egypt show the core constraining factors to lie in limited human and organizational capacities [26].

7.4 Cultural preservation versus social disruption

An information society Should both benefit from and contribute to worldwide cultural assets. A balance should be maintained between import of artistic, recreational, and educational products and local generation and consumption of culture-enhancing products. Without that balance, the information revolution could weaken the cultural ties that bind communities and nations. The information revolution greatly facilitates production, publishing, and international dissemination of cultural products by individuals, citizen groups, and communities.

Enhanced access to telecommunications and broadcasting creates opportunities for those who would enhance culture, society, and human life, as well as for those who would diminish or destroy them. Social control through information censorship is unworkable, yet information technology creates threats to individual privacy. Advocates of a benevolent and fair society must accept new responsibilities for influencing access to communications.

7.5 Social and employment inequities

Information technology has an increasing capacity to capture the knowledge and expertise required for many human tasks and to repeat those tasks with an unprecedented degree of speed and precision. This capacity constitutes a singularly powerful force for transformation of the employment landscape for both high-skilled and low-skilled jobs.

Many low-skilled jobs will be lost. Others will migrate to the lowest bidder Countries can compete for the migrating jobs, but they must mainly count on massive retraining to raise the skill level of the workforce as a whole. High-skilled jobs will increase their wage differential, but the information revolution will reduce the number of jobs in this category as human skills are increasingly automated. Furthermore, a shrinking pool of highly effective suppliers who best master technology and knowledge for competitive advantage will perform many of the remaining jobs.

The transferring of investment and operation responsibility for the telecommunications infrastructure from the public to the private sector is an urgent and particularly complex task. It involves sacrificing local short-term revenues and control to gain the investment volume and service improvements that a competitive private sector can supply.

From 1989 to 1994, the Ministry of Finance of Kenya built a public financial accounting and budgeting system. Creation of this system ran into difficulty in recruiting technical staff- a problem typical of developing countries. A training program was developed to train computer programmers, develop, their organizational skills, and instill in them the importance of accountability. As team-work is uncommon in Sub-Saharan Africa, the training emphasized the critical role of teams in developing and maintaining complex systems. The six-month programs produced 20 programmers at a cost of appoximately US $3,000 per person. Turnover rates, however, stood at about 50% [24]. Japan is just beginning to go online and few companies have taken the information technology route. It is estimated that just 8.6% of all personal computers in Japan are hooked into a network of some kind, compared with 52% in the US. This may be partly due to different attitudes on information sharing. The US is far more comfortable with putting information out- giving access to anonymous people and being non- hierarchical. That's not customary in Japan. To address this problem, Nippon Telegraph & Telephone and researchers at Stanford University's US- Japan Technology Management Center co-sponsored Japan Window, an electronic information service offering Internet users access to information on Japanese technology, science, business, industry, government and culture and allowing Japan to learn about the Internet by developing software products [25].

7.3 Waste and misallocation of scarce resources

In the absence of appropriate policy incentives, adequate quality stanards, and competitive discipline, a society could allocate scarce development resources to information infrastructure investments that create waste or perversely increase social inequality. Such is the case when telecommunications services are available only to the urban rich, when incentives favor use of technology more for recreational than productive purposes, and when an excessive share of investments are directed toward military purposes.

Wasteful information infratructure investments also can occur becausè of inadequate definition of the expected benefits and inadequate measurement of the actual results. Benefits in terms of societal capabilities and institutional objectives can be and should be defined for projects, but it is very complex to measure the results objectively, and it is tempting to hide behind alleged intangible benefits.

6.7 Creation of consortia for advice, investment, and research

Social networks and consortia have systematically addressed market failures and externalities in the supply and diffusion of information technology through financing of prototypes and large investment projects, advising on intangible investments related to physical investments, and sharing of best practices among enterprises. These are enabling conditions for program impact and wide diffusion of information technology and its benefits.

Science networks are supplanting journals. A daily wire service is operational for preprints and live reports by researchers in more than 10 disciplines in science and mathematics. About 20.000 e-mails flow daily to more than 60 countries with abstracts of new preprints and research reports. Thousands of full papers are downloaded daily [23].

Printed scientific information has been doubling every 12 years. Journal subscrption prices doubled in 8 years. A 1994 directory lists 440 electronic journals and research newsletters, up from 110 in 1991. Proponents and some agreements are emerging on electronic low-cost dissemination of scholarly work, but not yet agreement on refereeing. scholarly priority, journal of record, and other issues traditionally resolved by print journals. Scholarly journals could become an anachronism. In this change is an opportunity for Egypt and other developing countries: Participation in international research will depend on ideas, results, and evaluations by peers, not on location.

7. INFORMATION SOCIETY CHALLENGES

7.1 Constraints and hazards that must be recognized

The information revolution fuels change of remarkable pace and scope, but also of remarkable unevenness. Failures and horror stories litter the history of the information industry. Even the success stories, when scrutinized, reveal formidable social and technical complexities. The power of information technology, which has so much potential for social good, can also be harnessed for selfish, dangerous, or even destructive ends.

7.2 Social and technical complexity

Developing and deploying information systems and telecommunications capabilities is socially and technically complex, even in the face of technoligical advances and obvious potential benefits. Countries need to marshal substantial resources, often from abroad, to accomplish this task. They need to establish effective incentives and management schemes to facilitate adoption and effective use of new systems.

A World Bank report on the East Asian miracle discusses education: "Investments in both physical capital [for education] and schooling contribute significantly to economic growth. An increase of 10% in the primary or secondary school enrollment rate would raise the per capita income growth by 0.3%. Primary education is by far the largest single contributor to the HPAEs' [High Performance Asian Economies] predicted growth rates. Between 58% (Japan) and 87% (Thailand) of predicted growth is due to primary school enrollment. Physical investment comes second (between 35% and 49%), followed by secondary school enrollment. Far and away the major difference in predicted growth rates between HPAEs and Sub-Saharan Africa derives from variations in primary school enrollment rates. Education is the main theme of the story of the difference in growth between Sub-Saharan Africa and the East Asia high performers [22].

6.6 Developing of sectoral information networks

Social networks based on electronic communications are needed to enable alliances that connect institutions in agriculture, education, health, banking, industry, and other sectors. Once operational, these networks multiply opportunities for technical cooperation, research, coordination, information sharing, and private transactions.

Electronic customs clearance, transpotation arrangements, and other electronic commerce capabilities have reduced import and export transaction costs substantially for many countries. More than 60 commercial, educational, non-profit, and government entities in the U.S. have joined CommerceNet, formed in April 1994 to facilitate the use of an Internet-based infrastructure for electronic commerce to allow efficient interactions among customers, suppliers, and development partners to speed time to market and reduce the costs of doing business.

Toolnet is a network for small scale development projects that fosters exchange of information, experiences, expertise, and solutions to technical problems. It provides multifunctional electronic mail to link field workers, local organizations, technological institutions, international development organizations, and individuals to each other and to national and international networks. It is sponsored by TOOL, a non-profit organization in the Netherlands directed toward technology tranfer to and among developing countries. Toolnet Access Points are operating or planned in about 25 countries worldwide [21].

The Minitel network in France connects nearly 30 million private and business subscribers in France through 6 milion small terminals operating through the normal telephone network. It carries about 15.000 differnet services, which have created around 75.000 net jobs. It has indirectly helped to create 300,000 new jobs boosting efficiency and competitiveness.

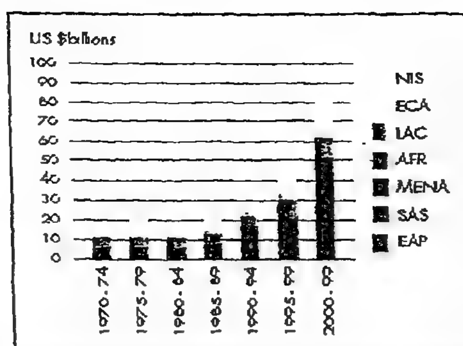


Figure 2: Average Annual Telecom Investment in Developing Countries

6.4 Strategic information systems for Egypt

Should include sector-wide information systems for education, health public sector management, and transportation. Electronic payments, education and science networks, trade facilitation, disaster prevention and management, and national statistics might also be defined as strategic systems.

An information system provides a societal capability based on the use of information that encompasses its full context of people, institutions, policies, processes, incentives, data, information technology, and infrastructure. A strategic information system provides a fundamental capability of such importance that it can enhance the scope and efficiency of an entire sector and economy.

6.5 Meeting new demands for education and training

Demand for specialized information professionals, computer, computer literacy throughout the workforce, and lifelong training are challenges for most countries in the face of rapidly changing technology and shifts in job mixes. Meeting the demand will require the efforts of universities, private companies, training institutions, computer societies, and accreditation councils.

Effectiveness of telecommunications technologies for secondary and post-secondary education has been fairly well documented. Research on the costs and benefits is much more limited. Hewlett Packard provides interactive voice and data communications in more than 100 classrooms worldwide, and estimates that its system delivers training at one-half the cost of traditional classes [17].

Some 10.000 schools in the US have a networked computer Lab. Increasingly, they are linked nationally and internationally. National Geographic and other have used networks to conduct experiments and aggregate data [21].

estimated call revenues exceed incremental cost of supply and (2) public payphones [19].

The emergence of the global information infrastructure presents an extraordinary opportunity for Egypt and other developing countries, sometimes called telecom lite. The new infrastructure is characterized by a variety of low-cost options for local connections, competing global operators for long-distance services, digital transmission, and low-cost, reliable, simple network access with increasingly sophisticated terminals. The model for the telecommunications infrastructure allows developing countries to leapfrog the industrial world's enormous investment in wired local loops, the most expensive part. Egypt can provide better service at a much lower cost per subsriber. With radio and satellite options for the local loop, widespread deployment of telecommunications becomes affordable, and Egypt can have the technical support to coexist with commerce in the information age [20].

The growth in telecom lines in developing countries in shown in figure 1, and in figure 2 the average annual telecom investment in developing countries is shown [9]. Telecommunications investment in less developed countries has been and will be massive. Regional codes are as follows: NIS Newly Industrialized States of former Soviet Union, ECA Eastern Europe and Central Asia, LAC Latin America and Caribbean, AFR Sub-Saharan Africa, MENA Middle East and North Africa, SAS South Asia, EAP East Asia and Pacific. Data for charts from [9, 10].

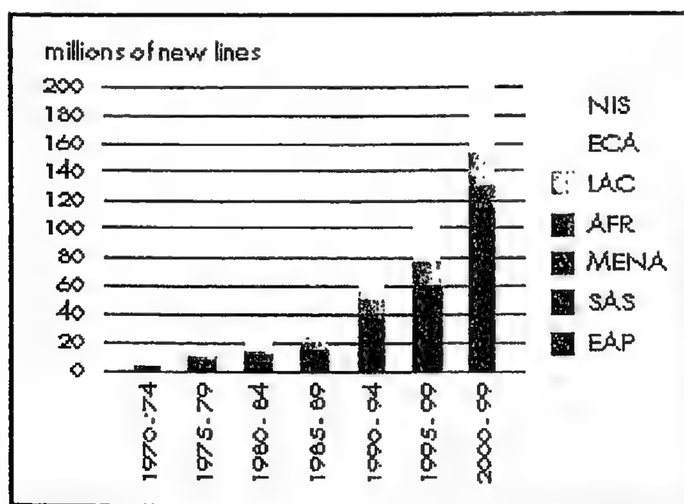


Figure 1: Growth in Telecom Lines in Developing Countries

6.3 A national information infrastructure

Consists of both the telecommunications networks and the strategic information systems necessary for widespread access to communicatiôns and information services. Deployment of this infratructure is a task of ten years or more and billions of dollars in Egypt and most developing countries. It is possible only through the efficient functioning of national and international markets for financing, engineeing, and management services. An information one friendly environment is a precondition for investments. While the role of the private sector is paramount, the government and often international organizations such as the World Bank Group also play key roles as legislators, regulators, advisors and guarantors.

There is a correlation between telecommunications investment and economic growth. A study by Hardy of 60 nations from 1960 to 1973 showed a causal relationship in both directions. Of course telecommunications investment increases as an economy grows, but there is also a small but significant contribution of telecommunications to economic growth. A causal relationship did not hold for radios. It was therefore postulated that the organizational impact of telecommunications contributed to economic growth. The study also found that the economic impact was greatest in countries with low teledensity (telephone lines per 100 persons) [17].

Rural telephone switches are being produced by and for India. In 1980, India had fewer than 2,500,00 telephones. almost all of them in a handful of urban centers. The country had only 12,000 public telephones for 700,000,000 people, and 97% of India's 600,000 villages had no telephones at all. India was using its priceless foreign exchange to buy the West's abandoned technology and install obsolete equipment. The technological disparity was getting bigger, not smaller. By 1987, a three-year effort by many of India's great young engineers had produced a series of telephone switches, manufactured in India to international standards, and adapted to village use. The 128-line rural exchange was housed in a metal container, cost about $8.000, required no air conditioning, could be installed in a protected space in the village, and would switch phone calls more or less indefintely in heat, dust, and monsoons [18].

The costs of telecommunications are falling, according to desk research and case studies from 26 developing countries that explores options for commercial supply of rural telecommunications. Capital costs per line for the most appropriate technology remains high but are falling: $1000 near urban areas and $2000-3000 in rural areas. This kind of service can be commercially feasible with annual revenues per line around $400 (urban) or $800-1200 (rural). The proposed strategy for serving villages and rural population centers, which is derived from a broad consensus of world experience, is to provide lines to : (1) institutions and businesses whose

There is a real and present danger of deepening inequity of access to information both within and among countries. The information revolution confronts countries with a major new equity issue, not of wealth but of information. If the adjustment agenda does not lead a country toward widespread access to information services, those without access will face a double hardship, lacking both information and economic opportunity.

6.1 Strategic consensus among partners.

Government, the private sector, local communities, non-governmental organizations, and international organizations such as the World Bank Group all have a role play in creating information economies in Egypt. The broader and more effective the partnerships among these agents, the faster the adjustment will accur. The most important role of partnerships is to create consensus on strategic priorities and commitment by the various stakeholders to a national strategy to create or enhance information infrastructure in Egypt. A national strategy must include identification of the sectoral reform policies and strategic investments needed to deploy the infrastructure as well as priorities for implementation. Reaching strategic consensus is a difficult but very important initial step toward an information economy.

Government action and reform can build competition. Private sector competition fueling growth of telecommunications and information industries could not have taken place in advanced countries without complementary government action and reform. Nor could have development of national infrastructures, without public investment in certain critical areas (such as R & D) or changes in laws and regulations.

Asian informatics strategies have public and private support. Taiwan identified production of PC clones as a strategic industry. Japan's Sigma project to expand software tool usage and common development platforms is a public/private partnership. Singapore's trade system initiative into an informatics strategy [14, 15].

6.2 Creation of an information-friendly environment

Characterized by coherent telecommunications reform and information policies; laws protecting investment, intellectual property, and individual privacy; open and well-regulated information and communication markets; education policies that favor a skilled labor force; and effective regulatory and standard-setting institutions. Such an environment supports availability, diversity, and low cost of information services and products. Clearly, the task of creating such environment falls to not but all of the partners for development [16].

applications. Export of high-skill jobs requires advanced and lifelong training. Information systems have supporting roles in education-efficient administration, low-cost delivery, and production of appropriate educational materials. Computer-based training is an effective tool for lifelong learning.

Education correlates with employment, income, and opportunity. In industrial countries, the well-educated are more likely to be employed. In the US in 1989, unemployment was 9.1% for persons with high school or less education and 2.2% for those with college degrees. In Japan, these figures were 7% and 2.3%. The well-educated earn more, and the gap is widening. In the US in 1980 the earnings gap was 30%; in 1988 it was 86%. The well-educated land jobs that provide them with more training; the uneducated are locked out of opportunities to improve their skills [11].

Distance education in Africa has been used to pursue entirely conventional educational ends. Its main advantages: economy, flexibility, and suitability for widely scattered student bodies [12]. Egypt is paving the way towards long distance education as an effective means of providing quality education.

The Globe program is a good example of distance education. It is a world wide science and education program coordinating the work of students, teachers, and scienists to study and understand the global environment. Globe is an international partnership involving countries around the world to: increase environmental awareness, contribute to scientific understanding of the Earth, and help students everywhere reach higher standards in science and mathematics [13].

6. NEEDS AND MEANS TO IMPLEMENT A MAJOR ADJUSTMENT AGENDA

It is easy to get excited about the opportunities, but the information revolution dictates new needs for Egypt that must be addressed, whether exciting or not. Fortunately the information revolution also provides the means to address this urgent agenda of adjustments.

Egypt and other developing countries will pay a high cost if lag behind. Without fast and successful adjustment, a software industry in Egypt can be hampered in many ways: Local software production may be less competitive in international markets against better quality, lower-priced, and more flexible products, especially one's that come from India, Malaysia, Philippines, Chian, Colombia, Korea and Singapore. Productive software firms might lose market share due simply to inability to connect to international trading networks and to participate on equal terms in markets. Research and development might stagnate without convenient access to international science networks.

population receptive to technological change. As economies become more service-intensive, workers must be retrained more frequently, and their performance becomes more dependent on access to IT. Accordingly, the diffusion of computer literacy shoud receive special attention in education strategy which in already the case in Egypt.

In sum, most dynamic trade routes of the twenty-first century will be dominated by transactions in intangibles rather than goods. Service industries will be responsible for the "roads" of the global "infostructure" and they will be the main providers of the content to be traded via electronic means. The adoption of a liberal trade and investment regime is essntial for countries to maximize the benefits to be derived from the internationalization of services and to move toward the information age. This is particularly true for developing countries especially Egypt.

4.3 Information Technology : The cutting Edge of the Services Revolution

The convergence of computer and communication technologies is promoting the development of computer-mediated (or electronic) networks. These networks are formed by systems of computers and communication hardware, and software that allows users to communicate and transmit data and other types of information. As digital switches -- computers replace electromechanical switches, and integrated services digital networks (ISDNs) emerge, the technical feasibility of new value-added services expands rapidly. Falling prices for hardware and software generate a positive feedback effect as the demand for network services is affected not only by prices but also by the expected size of the network [9,10].

This expansion is expected to accelerate as the cost of communication bandwidth (the capacity to transmit more bits of information simultaneously) continues to fall. Moreover, the cost of communication is becoming independent of distance, and networks are becoming more international. Transnational corportions, for example, are actively building international dedicated networks to address their communication needs. Egypt is paying great attention for the expansion and enhancement of the communication infrastructure [9].

5. ROLE OF EDUCATATION AND LIFELOGN LEARNING

Education pays rewards to an individual for an educational investment include more employment options, higher income, and better future prospects. Education is the best route to improved status for work force worldwide. The need for continued education and training persists throughout a person's career. Export of low-skill jobs requires primary education of good quality, including language skills. India's large English-scientific and technical pool, give them the capability to team up with American firms to develop new products and processes for world-wide commercial

mode of supply) renders it difficult to make a comprehensive evaluation of the economic value of the offers and their liberalizing impact. In terms of industry coverage, developing countries covered a smaller subset of service activities in their offers than industrial countries. Tourism and travel-related services were the only activities in which a substantial number of developing countries made commitments. Commitments in the area of communication services -- an area of critical relevance for countries interested in pursuing outward-oriented strategies of development were quite limited. These commitments are mostly related to value-added telecom services (e.g., data processing, electronic data interchange) but cover less than 20 percent, on average, of the service activities negotiated under this category.

The liberalization of trade in services actually achieved under the Uruguay Round seems rather limited at present. However, while the immediate liberalization may be limited, the agreement paves the way for future multilateral liberalization. The framework agreed provides for continued negotiations to be completed over a two-year period, and nothing constrains members from undertaking further unilateral liberalization, provided it is consistent with the multilateral disciplines established by the GATS [8].

4.2 Other supportive policies.

The services revolution places a premium on the development of a competitive telecommunications system. Most developing countries, are hard pressed to meet the demand for even basic telecommunication services. and investment in networks for value- added services may be considered an unaffordable luxury. However, technology now allows a country to develop a dual structure for telecommunication services: a country can invest in low-cost, dedicated networks for business needs in parallel with expanding the basic infrastructure. The private sector can play a leading role in this process, as it has been applied recently in Egypt.

Providing access to modern, high-quality communication services is not enough. Countries can be at a competitive disadvantage in long-distance exports because of non-competitive pricing of telecommunication services. (This has been the case for Eygpt and Eastern Caribbean countries). The use of alternative means of telecommunications (e.g., low-cost satellite stations) may be inhibited by monopolistic practices of the basic telecmmunications providers. Establishing a competitive framework for the provision of telecommunication services is therefore necessary.

Another important constraint faced by developing countries concerns the quality and relevance of the training of their work forces. In-house training can partially mitigate the shortcomings of the formal educational system in preparing workers to use IT in service industries. The main challenge, however, is to make the general

Border policies account for only some of the impediments to internationalization. Services are regulation-prone, and the domestic regulatory environment can create additional barriers to international competition (state monoplies in service industries, legal barriers to entry in economic activities, price controls). Domestic deregulation is often a necessary complement to the opening up of the foreign trade and investment regime. Also, differences in regulatory environments for service industries across countries may restrict access on a de facto basis (for example, different standards for accreditation of professionals). Accordinggly, effective liberalization may also require harmonization of regulatory practices among major trading partners (e.g., as pursued in the context of the Single Market initiative in Europe).

Alongside unilateral liberalization of services, countries are pursuing liberalization though reciprocal negotiations. An important achieverment of the Uruguay Round is the adoption of the General Agreement on Tarde in Services (GATS), which extends multilateral rules and disciplines to services. Several recent regional integration arrangements have also included liberalization of services.

The GATS covers four modes of international delivery of services: cross-border supply (e.g., transborder data flows, transportaion services); commercial presence (e.g., provision of services abroad through FDI or representative offices and branches); consumption abroad (e.g., tourism); and movement of personnel (e.g., entry and temporary stay of foreign consultants). It broadly follows the GATT (General Agreement on Tariffs and Trade) tradition, emphasizing nondiscrimination [Most-Favored-Nation (MFN) and national treatment] and prohibiting policy instruments that resemble quantitative restrictions. It innovates, however, in covering transactions associated with commercial presence (that is, establishment trade) and introducing a concept of market access that encompasses nonborder restrictions (e.g., limitations on the type of legal organization under which foreign providers can operate are, in principle, prohibited).

Unconditional MFN is a basic obligation of signatories, but MFN exemptions are allowed. Market access and national treatment, in turn, are specific obligations under the GATS. They apply only to the service industries and activities specifically listed by the country in its schedule of commitments, at the level of each mode of supply and subject to the limitations made explicit in the offer. The GATS adopts a positive list approach with respect to sectoral coverage of service industries -- that is, only the industries scheduled in the offers of the negotiating parties are subject to GATS discipline. This practice is less transparent than the negative list approach adopted, for example, in the North American Free Trade Agreement (NAFTA), in which all service industries are covered unless specifically exempted.

The complexity of the agreement (with offers made by service activity and

It is important to note that markets for these services are sensitive to technological change. Long-distance services in data entry, for example, are expected to continue to expand in the near future, reflecting the continuous fall in communcation costs. Progress in optical recognition technology and the development of online services for credit card and check clearing, however, can significantly affect the need for data entry in the future. These services may lose some of their dynamism as they are displaced by innovations in software and scanner technology in the industrial world. Nonetheless, the increasing number and diversity of information-intensive jobs, the technical feasibility of new long-distance services (e.g., in remote clerical support), and the dynamism of Foreign Direct Investment (FDI) flows and of the global demand for software suggest that the overall market for long-dictance services will continue expand [5].

While creating possibilities for new exports, the internationalization of services is important also for Egypt as importers of services, Long-distance access to the "floating pool" of nonproprietary knowledge, for example, is being revolutionized by computer-mediated networks, such as the Internet. Electronic bulletin boards are becoming more sophisticated and increasingly effective as instruments for the transference of knowledge and for technical assistance. They can now combine text, voice, images, and video and their use may significantly alter the prospects for human capital assumulation in Egypt and other developing countries in the next few years [5].

4. CAPTURING THE OPPORTUNITIES

To capture the opportunities offered by the internationatization of services, Egypt will need to adapt their regulatory environments and develop supportive physical and human infrstructure [4,5,6,7].

4.1 Liberalization and regulatory reform.

Liberalizing the import regime for services is central to achieving increased efficiency and competitiveness in the provision of services. It allows businesses to import services that are not produced domestically or that are not available at a price and quality required for competitiveness. Liberalization also fosters efficiency by increasing competitive pressures on domestic producers of services. Because of the nonstorability of many services, Foreign Direct Investment (FDI) is the major mode of international delivery of services. Lowering barriers to FDI, therefore, is crucial. Increasing recognition by Egypt of the need for such reform is reflected in the liberalization packages encompassing the services sector that many of them have unilaterally initiated in recent years. Yet most service activities continue to face a more restrictive regulatory regime than do goods.

3. OPPORTUNITIES IN LONG-DISTANCE SERVICES

As service industries rely increasingly on IT, they tend to become more dependent on capital and human-capital inputs. This has led some analysts to suggest that developing countries like Egypt cannot compete internationally in services and that policies to liberalize trade in services would limited interest to them. This view is mistaken. Developing countries are already carying out areas of comparative advantage in IT- based services, a process that will continue to evolve. Moreover, liberalization is not only about expanding exports; even more important is its role in helping domestic producers gain access to more efficient and diversified services in world markets [4].

There remains much scope for expansion in Egypt traditional service export areas (e.g., tourism). A new area of special promise is long- distance services. Data entry was one of the first service activities to be internationally outsourced. This type of activity requires only a low level of computer literacy and limited interaction between the customer and the supplier. The customer mails paper-based data forms or sends scanned images of data forms electronically to the foreign provider for processing. The supplier sends the computerized data back via telecommunication lines or by mailing magnetic tapes. Countries in the Caribbean have been quite active in exploring the market for offshore data entry.

Software programming is another activity that is increasingly traded across borders, with subsidiaries or partners overseas entrusted with developing software that is transmitted electronically back to the parent company. For example, many leading international computer and software companies have set up R & D and production operations in Bangalore, India. The Indian software industry, which is growing rapidly, generated revenues totaling more than $500 million in 1993-94, two-thirds of which came from exports. It it estimated that India has captured roughly 12 percent of the international market for customized software [4].

There are no precise estimates of the size of the market for long-distance services that can be captured by Egypt. The fact that a significant share of these transactions take place at the intrafirm level clouds the picture. However, rough estimates suggest that 15 percent of the employment in services in industrial countries may be internationally by developing countries. The potential impact of the globalization of services in terms of job displacement in industrial countries does not seem very large. But, from the perspective of developing countries, the potential impact in terms of higher exports over the long term is significant, possibly as large as their current total exports of commercial services. There are important niches in the market for long-distance services that can be successfully exploited by Egypt with a literate work force and a modern telecommunications system.

crtical system. It's often not clear to the customer whether the software developers will understand the nuances of his or her requirements, nor is it clear how well developers can respond to ongoing changes in those requirement.

The second stage, however, isn't an all-nothing proposition: It work in some instances, not in other. It's particularly good for certain kinds of systems programming projects, where the interface and the end results are well understood and clearly defined.

2.3 Stage 3 : Offfshore Generic Software, i.e. Package Exports

The third stage in the growth of the offshore software industry is the development of generic software products such as word processing packages, spreadsheets, and database programs, generic applications that can be marketed both in the home country and in the rest of the world. This stage often begins with the Egyptian software developers producing software for their local markets. Stage 3 becomes more interesting when the software developer in Egypt decide to bring their software to the North American/European marketplace. Obviously, the same kind of problem can exist when dealing with any other market that already has an advanced, computer-literate user population. Even if the new software is bug-free, the developer's cultural assumptions and trappings may cause problems.

But the real problem involves marketing. The cost to bring even the smallest PC application into the American marketplace for example, can be daunting enough for the Egyptian start-up software company.

2.4 Stage 4 : Exploiting Home-Grown Expertise

The fourth and final stage in the development of a software industry involves the development of product that take full advantage of the native country's unique areas of applications expertise. There are many unique applications that originated in Egypt, all of them involve indigenous applications expertise that is completely independent of software technology per se, and all are being packaged as commercially salable product. Egypt has built some wonderful multimedia applications to catalog. manage, and display its vast treasure of historical artifacts [3]. These applications are likely to be far more sophisticated than anything Americans would build in the U.S.

Ironically, Eygpt has also built some of the world's most sophisticated financial applications for manging its vast array of loans and financial-aid packages from numerous countries, international agencies, and relief organizations. Other third-world countries might find such applications quite useful, and they certainly won't find anything like them in the cataloge of Borland, Apple, or Microsoft.

cost of a comparable American/European work force. In some cases, cost is not as important as availabilty. If the project involves an advanced technology (e.g. using Visual C++ for a client/ server application), it may be difficult for the client to find available talent in the normal labor pool.

There is also issue of stability. Offshore programmers are less likely to quit and take another job in the middle of a critical project. Most of the problems associated with the first stage of development within the software industry are obvious: The overhead expenses reduce the competitive advantage of lower salaries, without providing any benefit to the client or the software vendors. Visas and work permits can involve enormous red tape, and the delays may be unacceptable to a client with a tight deadline. Language problems, cultural differences, and the difficulty of trying to maintain a live-style comparable to the higher-paid U.S/ European colleagues with whom the foreign programmer is working. All these are obvious but predictable problems. Similarly, the client will often express strong hesitation about hiring a foreign programmer without an on-site interview. This poses a financial risk for either the Egyptian vendor or the client, depending on who pays the travel costs.

A more subtle problem involves the "brain drain" from Egypt. In the long run, perhaps the U.S./Europe should be rejoice in the fact that its software industry is becoming a "melting pot", just as its manufacturing industries did throughout the nineteenth century. But national planners in the Egypt and other developing countries are very concerned, they are losing their best and brightest mind, because many of the programmers who go to the U.S./Europe for software project never come home. (Ironically, Egyptian government seems actually encouraged this practice in other fields, because expatriate workers typically send home a steady stream of cash to their relatives. This was judged to be a more effective way of bringing hard currency into the country than such traditional mechanisms as building a factory).

2.2 Stage 2 : On- site Analysis, Offshore Code

The second stage involves having a small group of systems analysts who will work with the customer, on-site, to define the system requirements, which they then transmit back to programmer in the home country. Most of the first-stage offshore players are planning to move in this direction. Using a small number of on-site analysts saves considerable transportation and overhead costs. Having home-country software engineers implement the design also minimizes the brain-drain problem and allows the Egyptian firms to begin buiding its own infrastructure and long-term expertise in software technologies. The problem and the disadvantages of the second developmenttal stage are obvious: End users are reluclant to trust an offshore firm several thousand miles away with the development of a mission-

vation and quality, on the whole Egyptian software companies are not entrenched in the world-wide arena. In fact, due to relatively small local Egyptian market, it is essential for Egyptian software companies to effectively reach export market, in order to achieve growth and acquire market share. While many of Egyptian smaller software companies are still focusing on local markets, export markets typically account for well over 60% of product sales among Egyptian larger and most successful software companies, like Delta.

While a world-wide presence is the ultimate goal, the US market represents the most immediate and logical opportunity. Barriers related to language, trade, and standards can be overcomed compared, for example, to Europe and Asia. Furthermore, this market not only repersents enormous revenue generating potential, it has a voice that is heard throughout the world. Products that are successful in the US typically have more chance to succeed elsewhere. Market acceptance of a product in the US means almost instant recognition worldwide. The strategic importance of the US market is fully understood in the Egyptian industry, but what is less obvious is how to succeed and continue to succeed in it. One major challenge is to have in place effective distribution channels.

Small and medium-sized Egyptian software companies need useful input to their marketing and distribution strategies, by identifying the trends affecting the distribution of software products in the United State and Worldwide, and assessing their impact on the use and effectiveness of distribution channels, developing strategies for success, and practical conclusions and recommendations. The results are based upon in-depth secondary research, interviews with representatives of software companies, channel organizations and industry associations in Egypt, US, and Europe as well as analysis by industry experts at Egyptian Software Association (ESA) & and Egyptian Society for Information System And Computer Technology (ESISACT).

2. PATH TO INTERTIONAL SOFTWARE DEVELOPMENT

Egypt is expected to take a clear path with regard to international Software development. This path has four distinct stages [2]. and each stage offers some interesting opportunities.

2.1 Stage 1 : Export of Labour, i.e. Live in Contract Programmers

At first, teams of Egypt programmers will be knocking on doors, offering to develop software with teams located at the customer's local site. The primary advantage to this approach is low price. Even with the overhead of travel, lodging, and administrative expenses, the client often finds that he or she is paying only half the

The software industry is becoming an arena of worldwide competition. Measured by revenues, by the number of firms engaged in software development, and by the variety of available software, industry performance over the last six years has been extraordinary. Although data are lacking and not very reliable, it is estimated that the world market for software is increaing by 30 to 40 percent annually and could reach US $ 500 billion in 1999 [1].

Five major forces are now promoting rapid changes in the worldwide software industry: (a) A severe software personnel constraint, with software and software related support activities now accouning for the overwhelming percentage of total system costs; (b) The global battle for operating system standards; (c) The move away from single vendor solutions as the typical way for organizations to meet their information systems needs toward customized, integrated, multi-vendor hardware and software solutions; (d) The increasing emphasis on software production and sales by hardware vendors, leading to increasing concentration within the industry of large and medium-sized firms; and at the same time (e) An expansion and fragmentation of the industry in growing number of independent software vendors.

This paper examines the global software market, particularly the major trends in the international software industry. The paper then reviews the software development process in Egypt and its labor, education, and training requirements. New technologies will affect the entire software production process and will require a reorientation in thinking about investment and technological options. New organizational and software management practices are also being introduced into software firms in an attempt to control costs throughout the software engineering process, and to improve the quality of the products and an organization's process of developing software. The major software engineering problems of the future will require organizational solutions that: integrate advanced software development tools based on object oriented technology and knowledge-based systems; provide flexible policies to ensure the best use, education and training on these tools and new methods; and manage large system development projects. Some areas of software production, especially for export markets, may be more cost effective than others. Ways in which Egypt can exploit some of these opportunities are addressed. Software firms in Egypt still have some way to go, and their role in the international software industry is likely to quite modest at best. Specific strategies for building a competitive software industry or enhancing the competitiveness of that industry are also addressed.

Egyptian software companies are gaining recognition throughout the Arab world as producer of high quality, leading edge products. Yet this recognition is still a far from what is deserved. Despite their proven excellence in software inno-

CHAPTER 1

SOFTWARE INDUSTRY IN EGYPT: OPPORTUNITIES AND CHALLENGES

Dr. Alaa Mohamed Fahmy

ABSTRACT

The role, status and future of software industry in Egypt, have been largely misunderstood by most of the western world. The ecnomic, political and strategic significance of Egypt make technological lacuna about it inexcusable. Egypt is the most populous country in the region as well as one of the most internationally oriented computing community in the Arab world, and its trained people are among its most important high-tech exports. Egyptian Government has been undertaking an ambitious steps towards economic reform and technology development. Actions have been taken to support software industry in Egypt. This paper highlights the software potential in Egypt, its role in providing high quality employment for young people, its innovative software packages and its ability to fulfill regional and offshore development and service contracts. Challenges facing the growh and globalization of Egyptian software Industry and the notion of the information society and information- friendly environment is also discussed.

1. INTRODUCTION

The worldwide computer industry has been characterzed since its inception by continuous innovation, improvement, and rapid change. These developments have resulted in the growing importance of software, the programs that run computers and allow them to communicate with each other through data networks. Developing and newly industrialized economies (NIEs) seeking a share of the burgeoning global information industry are promoting the development of their software industries through a variety of policy and institutional measures. At the same time, a growing number of U.S. and west European software and non- software firms are beginning to develop software overseas, often for foreign as well as domestic markets [1].

PART ONE

THE EGYPTIAN MARKET AND SOFTWARE INDUSTRY

Table of Contents

DEVELOPMENT OF SOFTWARE INDUSTRY IN EGYPT

Proceedings of

The Fifth Scientific Conference of Information Systems and Computer Technology

Cairo : 9 - 11 December 1997

Organized and Conducted by

The Egyptian Society for Information Systems and Computer Technology (ESISACT)

Editor
Prof. Dr. Mohamed M. El Hadi
President of ESISACT



The Publisher
ACADEMIC BOOKSHOP
2000

DEVELOPMENT OF SOFTWARE INDUSTRY IN EGYPT



DEVELOPMENT OF SOFTWARE INDUSTRY IN EGYPT

Proceedings of

The Fifth Scientific Conference of Information Systems and Computer Technology

Cairo : 9 - 11 December 1997

Organized and Conducted by

Editor
Prof. Dr. Mohamed M. El Hadi
President of ESISACT

ACADEMIC BOOKSHOP

